University of Alberta

Categorizing Digital Documents by Associating Content Features

by

Catalina Maria-Luiza Antonie

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Fall 2002

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-81356-8

Canada

University of Alberta

Library Release Form

Name of Author: Catalina Maria-Luiza Antonie

Title of Thesis: Categorizing Digital Documents by Associating Content Features

Degree: Master of Science

Year this Degree Granted: 2002

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Catalina Maria-Luiza Antonie
305-8516-99St
Edmonton, Alberta
Canada, T6E3T6

Date: 08/08/02

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Categorizing Digital Documents by Associating Content Features** submitted by Maria-Luiza Antonie in partial fulfillment of the requirements for the degree of **Master of Science**.

Dr. Osmar R. Zaïane

Dr. Marek Reformat

Dr. Walter F. Bischof

Date: 08/05/02

Abstract

The continuous expansion of the available data has triggered a lot of research fields, Data Mining being only one of them. Extensive research has been conducted in Data Mining community and many algorithms for classification, clustering, association mining, etc. are now available to deal with large amounts of data. In this work, we propose a new classification method and we report our tests on medical image and text document collections.

In this thesis we propose a new method to construct a classification system using a novel model. This model uses association rule mining to discover interesting patterns in data that can be further used in building a classifier. First, we model our input data into transactions, then our system generates the association rules from this transactional database using an apriori-like method. It uses this set of rules or a pruned one to classify new tuples. The classification is based on evaluating a multiple set of rules that match the new tuple to be classified. By using the confidence of the rules as well as a *dominance factor* a prediction is made.

Our approach has the advantage of a very fast training phase, and the rules of the classifier generated are easy to understand and manually tuneable. Our investigation leads to conclude that association rule mining is a good and promising strategy for efficient automatic classification.

Acknowledgements

I would like to take this opportunity to express my thanks toward everyone who supported me in completing this thesis. My gratitude goes to my supervisor, Dr. Osmar Zaïane, who introduced me to the Data Mining research. I am thankful for his support, guidance and thoughtful discussions. His help and supervision have been invaluable.

I would like to thank my family for helping me become the person I am. Without them I would not be here.

My gratitude goes to my husband for his continuous love and encouragement.

Dedication

To my parents and my husband

Contents

1 Introduction **1**
- 1.1 Our Contribution . . . 2
- 1.2 Thesis Organization . . . 3

2 Related Work **4**
- 2.1 Association Rule Mining . . . 4
  - 2.1.1 Association Rules . . . 4
  - 2.1.2 Generating Association Rules . . . 5
- 2.2 Classification . . . 7
  - 2.2.1 Associative Classifiers . . . 8
  - 2.2.2 Probabilistic Classifiers . . . 12
  - 2.2.3 Decision Trees . . . 13
  - 2.2.4 Decision Rules . . . 14
  - 2.2.5 Example-based Classifiers . . . 16
  - 2.2.6 Neural Networks . . . 17
  - 2.2.7 Support Vector Machines . . . 18
  - 2.2.8 Boosting and Bagging . . . 19
  - 2.2.9 The Rocchio Classifier . . . 20
  - 2.2.10 N-grams vs. Unigrams . . . 21
- 2.3 Document Categorization . . . 22
  - 2.3.1 Text Categorization . . . 22
  - 2.3.2 Image Classification . . . 23

3 Association Rule-based Classification **25**
- 3.1 Association Rule Generation . . . 26
  - 3.1.1 Association Rule-based Classification with All Categories 27
  - 3.1.2 Association Rule-based Classification by Category . . . 30
- 3.2 Pruning the Set of Association Rules . . . 33
  - 3.2.1 Removing Low Ranked Specialized Rules . . . 33
  - 3.2.2 Pruning Rules based on Database Coverage . . . 34
  - 3.2.3 Other pruning techniques . . . 35
- 3.3 Prediction of Classes Associated with New Objects . . . 36
- 3.4 Comparison between our approaches and the other existing associative classifiers . . . 39

4 Experimental Results: single label classification **43**
4.1 UCI Datasets 43
4.2 Mammogram Collection 45
4.2.1 Database Description 46
4.2.2 Preprocessing Techniques 47
4.2.3 Transactional Database Organization for ARC-AC . . . 49
4.2.4 Transactional Database Organization for ARC-BC . . . 50
4.2.5 Experimental Results 51
4.3 Gene Categorization 57

5 Experimental Results: multiple label classification **60**
5.1 Benchmark Collections 60
5.1.1 Text Corpora 60
5.2 Evaluation Measures 62
5.3 Experimental Results 64

6 Conclusions **69**
6.1 Thesis Summary 69
6.2 Conclusions 70
6.3 Future Work 70

Bibliography **72**

List of Figures

2.1 Decision Tree 13
2.2 N-Layer Neural Network 17
2.3 Support Vector Machine Classifier 19

3.1 Classifier for all categories 27
3.2 Classifier per category 31

4.1 Image categorization process 45
4.2 Pre-processing phase on an example image: (a) original image; (b) crop operation; (c) histogram equalisation 48
4.3 Mammography division 49
4.4 Success rate of ARC-AC with an equilibrated split 53
4.5 (a)Precision over the ten splits ; (b) Recall over the ten splits; 56

5.1 Classifier 62

List of Tables

2.1 Examples of decision rules 14
2.2 Classification methods comparison 21

3.1 A transactional database 27
3.2 1-itemsets from the transaction table 3.1 with their supports and possible correlations with the class labels 28
3.3 The Training set for C1 31
3.4 The Training set for C2 31
3.5 The Training set for C3 32
3.6 Comparison among the existing associative classifiers and the proposed methods in this thesis 42

4.1 The success rate comparison of ARC-AC, C4.5, CBA and CMAR 44
4.2 Success ratios for the 10 splits with the association rule based classifier with all categories (ARC-AC) 52
4.3 Classification accuracy over the 10 splits using ARC-BC . . . 54
4.4 Contingency table for category *cat* 56
4.5 Classification accuracy over the 10 splits using ARC-AC [3] . . 57

5.1 Statistics for ten most populated Reuters categories 64
5.2 Precision/Recall-breakeven point micro-averages for ARC-BC 65
5.3 Precision/Recall-breakeven point on ten most populated Reuters categories for ARC-BC and most known classifiers 65
5.4 Micro-average Precision/Recall-breakeven point for ten most populated Reuters categories with different classifiers 66
5.5 Examples of association rules composing the classifier. 66
5.6 Micro-average Precision/Recall-breakeven point for ten most populated Reuters categories - manual tuning of the classifier . 67
5.7 Training and testing time (in seconds) with respect to the support threshold for Reuters-21578 dataset 67
5.8 Precision/Recall-breakeven point for ten most populated Reuters categories with different pruning methods 68

Chapter 1

Introduction

In the last decade, with the continuous growth of the digital collections Knowledge Discovery and Data Mining have become a very important research field. Not only are interesting knowledge and patterns in databases discovered, but Knowledge Discovery in Data and Data Mining can deal with large amounts of data. There are great demands for the data mining algorithms when discovery of useful knowledge is needed. Data Mining algorithms can be used in various domains including market basket analysis, medical applications, business management, space exploration, financial data analysis, etc.

Classification and association rule analysis are two important problems in Data Mining. Classification is the process of finding a set of patterns that can distinguish among different classes in a data set for the purpose of using this knowledge to predict the class labels for new incoming objects associated to this data set. Association rule analysis is the task of discovering association rules that occur frequently together in a given data set. Association rules have been extensively used in market basket analysis.

Building of an automated classification system is a very important task nowadays with the large existing collections. Although, in many companies there exist specialized people performing manual classification, their job becomes more difficult and unfeasible with every day. Not only that this task is very time-consuming, but it is also subject to errors due to fatigue or other factors. Mainly, in our thesis we concentrated on text categorization and mammography classification. A text categorization system can be used in indexing

documents to assist information retrieval tasks as well as in classifying e-mails, memos or web pages in a yahoo-like manner. In addition, a mammogram classification system could be a very useful tool in hospitals, assisting physicians in taking decisions. Needless to say, automatic classification is essential.

1.1 Our Contribution

In this thesis we propose a new classification model, based on association rule mining. In the past years, associative classifiers have started to attract attention [37, 35]. An important advantage that these classification systems bring is that, using association rule mining they are able to examine several features at a time, while other state-of-the-art methods, like decision trees, consider that each feature is independent of one another. However, in real life applications, this assumption is not true, and it is proved that correlations and co-occurrence of features can be very important.

Although the associative classification systems presented in the literature started to attract attention, they still face some problems. None has tackled the problem of multi-label classification, which is very important to many domains, text categorization being just one of them. In addition, they only performed experiments on small datasets [47].

In this work we present a new classification method for text and image classification that takes advantage of association rule mining in the learning phase and makes the following contributions: First, *a new technique for text and image categorization that makes no assumption of term independence or feature independence is proposed. This method proves to perform as well as other methods or better than other methods in the literature.* Second, *the method we propose is fast during both training and categorization phases.* Third, *the classifier that is built using our approach can be read, understood and modified by humans.* Experiments show that the effectiveness of the classifier can be improved by manually fine tuning the classification rules generated during the training phase.

We tested our approach on various data collections: The motivation for

this choice was to prove the ability of our classification method to perform on different transactional databases, as well as to bring a contribution to text categorization and mammography classification fields.

1.2 Thesis Organization

The remainder of the thesis is organized as follows. Chapter 2 introduces the methods that are related to this work. Chapter 3 describes the associative classifiers that we propose. Chapter 4 presents some experimental results when our algorithms are employed for single label classification, while Chapter 5 discuss the results obtained when multiple label categorization is performed. The conclusions and future work are presented in Chapter 6.

Chapter 2

Related Work

The problem that we tackle is that of classification in a data mining context. Knowledge Discovery in Data deals with the discovery of interesting patterns in very large databases. This is a very important field nowadays due to the huge amounts of data that people have to deal with. That is why Knowledge Discovery and Data Mining have become such an important research field. Classification represents one of its important tasks.

In this chapter we review existing work in the literature that is relevant to the method that will be proposed in this thesis. There are two issues that merge in our work: classification task and association rule mining. The next sections will give more details on classification and association rule mining. Our model presented in the next chapter integrates classification with association rule mining. In addition, our system was tested in text categorization and medical image classification. Therefore, existing classification models in these domains are discussed as related work.

2.1 Association Rule Mining

2.1.1 Association Rules

Association rule mining is a data mining task that discovers relationships among items in a transactional database. Association rules have been extensively studied in the literature. The efficient discovery of such rules has been a major focus in the data mining research community. From the original *apriori* algorithm [2] there have been a remarkable number of variants

and improvements culminating with the publication on the FP-Tree growth algorithm [27].

Formally, association rules are defined as follows: Let $\mathcal{I} = \{i_1, i_2, ...i_m\}$ be a set of items. Let $\mathcal{D}$ be a set of transactions, where each transaction T is a set of items such that $T \subseteq \mathcal{I}$. Each transaction is associated with a unique identifier TID. A transaction T is said to contain X, a set of items in $\mathcal{I}$, if $X \subseteq T$. An *association rule* is an implication of the form "$X \Rightarrow Y$", where $X \subseteq \mathcal{I}, Y \subseteq \mathcal{I}$, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ has a *support* s in the transaction set $\mathcal{D}$ if $s\%$ of the transactions in $\mathcal{D}$ contain $X \cup Y$. In other words, the support of the rule is the probability that X and Y hold together among all the possible presented cases. It is said that the rule $X \Rightarrow Y$ holds in the transaction set $\mathcal{D}$ with *confidence* c if $c\%$ of transactions in $\mathcal{D}$ that contain X also contain Y. In other words, the confidence of the rule is the conditional probability that the consequent Y is true under the condition of the antecedent X. The problem of discovering all association rules from a set of transactions $\mathcal{D}$ consists of generating the rules that have a *support* and *confidence* greater than given thresholds. These rules are called *strong rules*.

2.1.2 Generating Association Rules

The efficient discovery of association rules has been a major focus in the data mining research community. Most popular algorithms, designed for the discovery of all types of association rules, are apriori-based. The next paragraph describes the apriori algorithm on which the models in our approaches are based.

The main idea behind the *apriori* algorithm is to scan the transactional database searching for k-itemsets (k items belonging to the set of items I). As the name of the algorithm suggests, it uses prior knowledge for discovering frequent itemsets in the database. The algorithm employs an iterative search and uses k-itemsets discovered to find (k+1)-itemsets. The frequent itemsets are those that have the support higher than a minimum threshold. Here is a formal description of the apriori algorithm as described in [28] :

```
(1)   scan the database and find 1-frequent itemset (S1)
(2)   for (k = 2, L_{k-1} ≠ ∅, k++){
(3)       C_k= apriori_gen (L_{k-1},min_sup)
(4)       foreach t ∈ D { //scan D for counts
(5)           C_t = subset(C_k, t)//get the subsets of t that are candidates
(6)           foreach c ∈ C_t
(7)               c.count++
(8)       }
(9)       L_k = {c ∈ C_k|c.count ≥ min_sup}
(10)  }
(11)  return L = ⋃_k L_k
```

procedure apriori_gen(L_{k-1},min_sup)

```
(1)   foreach itemset l_1 ∈ L_{k-1}
(2)       foreach itemset l_2 ∈ L_{k-1}
(3)           if (l_1[1] = l_2[1]) ∧... ∧ (l_1[k-2] = l_2[k-2]) ∧(l_1[k-1] ≤ l_2[k-1]) then{
(4)               c=l_1 ⋈ l_2 //join step
(5)               if has_infrequent_subset(c,L_{k-1}) then
(6)                   delete c // prune step
(7)               else add c to C_k
(8)           }
(9)   return C_k
```

procedure has_infrequent_subset(c,L_{k-1})

```
(1)   foreach (k-1) subset s of c
(2)       if s ∉ L_{k-1} then
(3)           return TRUE
(4)   return FALSE
```

The next step, after the frequent itemsets are generated, is to discover the strong association rules that can be found in the database. The discovery of the association rules is done as follows:

(1) **foreach** frequent itemset (S)

(2) generate all its non-empty sub-sets (NS)

(3) **foreach** of these subsets

(4) the rules that can be generated have the following form:
$NS \Rightarrow S - NS$

(5) **if** it exceeds the minimum confidence

(6) the rule is considered strong

While apriori needs k passes over the data for association generation, FP-tree [27] generates the complete set of frequent items without candidate generation. It stores the information about the frequent patterns in a tree structure and it mines all the frequent items by pattern fragment growth. FP-tree algorithm scans the database once to find the 1-itemsets. Then, it keep only those that have higher support than the given support threshold. It stores those frequent 1-itemsets in a list in support descending order. The next step is to scan the database again, in order to build the FP-tree which is a prefix tree with respect to the list found after the first scan. When the tree is constructed the entire set of association rules is generated.

2.2 Classification

The main idea of the classification task is to discover interesting patterns in training set of data that will be used later in the classification process. Classification has multiple applications and has already been applied in many areas such as text categorization, medical analysis, space exploration, etc. Although classification has a long history and there exist many popular techniques for classification, there is still room for improvement. Besides decision trees, Bayesian classifier, neural networks, support vector machines, the classification based on association rule mining started attracting attention in the past years [37, 35].

2.2.1 Associative Classifiers

Recently, in the literature, a new classification method has been proposed. In this case the learning method is represented by the association rule mining. The main idea behind this approach is to discover strong patterns that are associated with the class labels. The next step is to take advantage of these patterns such that a classifier is built and new objects are categorized in the proper classes.

Two such models were presented in the literature: CMAR (Classification based on Multiple Association Rules) [35] and CBA (Classification Based on Associations) [37]. Although both of them proved to be effective and achieved high accuracy on relatively small UCI datasets [47], they have some limitations. Both models perform only single-class classification (i.e. to each object is assigned a unique class label) and were not implemented for text categorization or image classification. In many applications, however, and in text categorization in particular, multiple-class classification (i.e. a document could be classified in many classes simultaneously) is required.

The main steps in building an associative classifier when a training set is given are the following:

1. *Generating the set of association rules from the training set* In this phase association rules of the form $setof features \Rightarrow classlabel$ are discovered by using a mining algorithm.

2. *Pruning the set of discovered rules* In the previous phase a large set of association rules can be generated especially when low support is given. That is why pruning techniques are a challenging task to discover the best set of rules that can cover the training set. This phase is employed to weed out those rules that may introduce errors or are overfitting in the classification stage.

3. *Classification phase* At this level a system that can make a prediction for a new object is built. The task here is how to make use of the set of rules from the previous phase to give a good prediction.

Classification Based on Associations (CBA)

Classification Based on Associations (CBA) [37] is the first associative classifier proposed in the literature. It integrates classification and association rule mining to develop a classifier based on association rules. It adopts an apriori-like algorithm [2] in the rule generation phase and then it employs some pruning techniques to reduce the set of association rules.

Generating the Set of Rules Generating the association rules is the first phase of CBA. It consists of finding all CARs (class association rules) that satisfy minimum support and confidence requirements. CBA finds all the items that exceed minimum support and they are of the following form: $< F, c >$, where F is a set of items and c is a class label. Each item that satisfies this format represents a rule: $R : F \rightarrow c$. If the confidence of the rule is greater that the minimum confidence the rule belongs to the CAR set.

Building the Classifier The authors chose to employ a pruning technique in order to select the best representative association rules from the CAR set. This pruning technique is based on the database coverage. CBA sorts all the rules in CAR according to the following definition.

Definition1 Given two rules R_1 and R_2, R_1 is higher ranked than R_2 if:

(1) R_1 has higher confidence than R_2

(2) if the confidences are equal supp(R_1) must exceed supp(R_2)

(3) both confidences and support are equal but R_1 has fewer attributes in left hand side than R_2

Then it selects the most representatives and high ranked rules according to the database coverage explained below:

(1) **foreach** rule R in the set CAR=sort(CAR)
(2) go over D and find those transactions that are covered by the rule R
(3) **if** R associates at least one transaction to its correct class
(4) select R
(5) remove those cases that were covered by R

(6) select a default rule (i.e. a rule that selects majority class in the remaining tuples in database)

The associative classifier consists of the set of rules that resulted when database coverage was applied.

Classification Phase In the classification phase, CBA searches the set of rules that represents the classifier starting with the highest rank. It finds the first rule that matches the new object to be classified. When such a rule is found, its class label is associated with the new object. Otherwise, the default rule is applied.

Classification based on Multiple Association Rules (CMAR)

Classification based on Multiple Association Rule (CMAR) [35] was proposed two years later. It outperforms CBA on the UCI datasets [47], on which both methods were tested. In the association rule mining phase it employs FP-tree growth algorithm [27]. It proposes a new structure for rule storage and employs pruning techniques for selecting a high quality set of rules. The classification process is based on multiple rule analysis (the decision is made based on the analysis performed on a set of rules).

Generating the Set of Rules Based on FP-tree growth algorithm in the rule generation phase, CMAR finds the complete set of rules in the form: $R : F \rightarrow c$, where F is a set of attributes and c is a class label. Only the rules that pass minimum support and minimum confidence thresholds are considered as candidates in building the classifier. The authors that introduced this method propose a new indexing structure for rule storage as well. This is a prefix tree structure, called CR-tree. It proves to be a compact structure and it is further used in the pruning phase which is detailed in the next paragraph. This phase is similar to the one in CBA [37] the only difference being the mining algorithm used. In addition, CMAR stores the rules in the CR-tree, structure which is used in pruning phase for efficiency.

Building the Classifier The classifier consists of a high quality set of rules that results after the pruning techniques are applied on the initial candidate set. CMAR employs three pruning techniques: specific rule elimination, selecting positively correlated rules and database coverage.

The rules are ordered according to the same ordering definition as in CBA. Then lower ranked specific rules are eliminated as described below.

Definition2 introduces the notion of specific rule and it is employed in pruning the low ranked specific rules.

Definition2 Being given two rules $T_1 \Rightarrow C$ and $T_2 \Rightarrow C$ we say that the first rule is more general rule w.r.t. second rule iff $T_1 \subseteq T_2$.

(1) sort the rules according to **Definition1**
(2) **foreach** rule in the set of candidate rules
(3) find all those rules that are more specific (according to **Definition2**)
(4) prune those that have lower confidence

The next pruning technique that is employed is selecting only those rules that are positively correlated. This is done by using the chi-square test.

The last pruning technique is database coverage. It is similar to the one used in CBA, except that for CMAR the authors introduced a coverage threshold δ. Each tuple in the database has a count associated with it. The count increases when a rule covers the tuple that is associated with the count. The tuples are eliminated only when the count is greater than the coverage threshold. In CBA the tuples would be eliminated when the count equals 1.

Classification Phase A set of rules was selected to form the classifier as discussed above. The classification phase in CMAR is based on multiple association rules. Consider a new tuple T to be classified. CMAR chooses the subset of rules that matches T. It divides the rules according to the class label and evaluates these groups based on chi-square analysis [35]. Then, it chooses the strongest group to classify T in a certain class. A pseudocode for this classification stage is given below.

(1) Select the rules that match T and put them in prediction-set P
(2) **if** all the rules in P indicate the same class label
    attach that label to the new tuple
(3) **else**
(4) divide P in subsets based on the class labels
(5) in each group compute the weighted chi-square for each rule
    and sum them up
(6) attach to T that class label that had the highest sum of
    weighted chi-square

2.2.2 Probabilistic Classifiers

Classifiers based on probabilistic models have been proposed starting with the first presented in the literature by Maron in 1961 [38] and continuing with naïve-Bayes [32] that proved to perform well.

Probabilistic classifiers estimate the probability of each class given the features of the new object to be classified. To determine these probabilities, Bayes theorem is used. Let $\mathbf{d}$ be a new instance to be classified. Its class is unknown. Let c_i be the hypothesis that the object falls into category c_i.

Bayes theorem is given by the following formula:

$$P(c_i|\mathbf{d}) = \frac{P(c_i) * P(\mathbf{d}|c_i)}{P(\mathbf{d})} \tag{2.1}$$

The above equation states that the object $\mathbf{d}$ falls in class c_i with probability $P(c_i|\mathbf{d})$. However, the computation of this probability is difficult. As a consequence, it is common to make an independence assumption that will make the computation easier. It is assumed that there are no dependence relationships among the object attributes. The independence assumption presumes that the attributes when considered as random variables are statistically independent.

Thus, equation 2.1 becomes:

$$P(c_i|\mathbf{d}) = \prod P(d_k|c_i) \tag{2.2}$$

Probability classifiers that make use of this assumption and compute the probability according to 2.2 are called Naïve Bayes classifiers. Despite the fact

that the assumption used is not entirely true in real applications Naïve Bayes classifiers are effective. More details and a thorough survey of the Bayesian classifiers is given in [32].

2.2.3 Decision Trees

A decision tree is a tree structure where each internal node represents a test on the attribute, each branch is labelled by the outcome of the test and each leaf node represents a category. An example of such tree is given in Figure 2.1. Let **d** be a new instance to be classified. In order to classify it, all its attribute values d_k are tested against the decision tree.

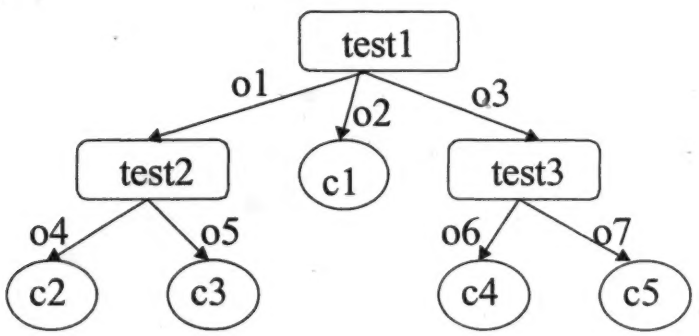


Figure 2.1: Decision Tree

A procedure for building a decision tree is to apply a greedy algorithm in a top-down recursive divide-and-conquer manner. The basic algorithm is as follows:

(1) create a node representing the training set
(2) **if** all the samples belong to the same category
(3) attach this label to the node, and it becomes a leaf
(4) **else**
(5) using a heuristic search find the term that best separates the samples into classes, and place each class in a subtree

The algorithm is used recursively to form a decision tree for each partition. The recursive process stops when:

(a) all samples for a given node, belong to the same class

(b) the samples cannot be further divided (no attributes left)

(c) there are no samples for a certain branch. In this case a leaf is created with the majority class.

Entropy is a common measure for how informative a node is in the tree. When a new tuple is presented to the system for classification, the attributes of this tuple are tested against the decision tree. Starting with the root node the attributes are tested until a leaf node is reached. At this point the value of the leaf node is attached to the tested tuple as class. There are some standard packages for creating a decision tree when a training set is given. Among the most popular are ID3 [25], C4.5 [41, 12, 13, 30] and C5 [36].

2.2.4 Decision Rules

Rule induction methods have been extensively discussed in the literature [41, 52]. A decision rule classifier consists of a set of rules in disjunctive normal form (DNF). The antecedent of the rule is formed by the presence or absence of terms in documents, while the consequent says whether to classify the object under class c_i or not. An example of such a set of rules is given in Table 2.1

$t_1 \bigwedge t_3 \bigwedge t_4 \Rightarrow c_i$
$\overline{t_2} \bigwedge t_5 \Rightarrow \overline{c_i}$
$\overline{t_1} \bigwedge t_2 \bigwedge t_7 \Rightarrow c_j$
$t_1 \bigwedge t_2 \bigwedge \overline{t_4} \Rightarrow \overline{c_j}$

Table 2.1: Examples of decision rules

The decision rules induction is done in a bottom-up fashion, while the induction in decision trees, as described above, is done in a top-down manner. When the building of classifier for category c_i starts, all the terms in each training document represents the antecedent of a rule. The consequent is either c_i, if the object is a positive example, or $\overline{c_i}$, if the object is a negative one. This process may lead to an overfitting set of rules. That is why, when a set of rules is inducted, a generalization and a pruning criteria are employed. The heuristics for these two steps differ from one classifier to another.

The inducing of a set of decision rules starts with an empty set of rules. The generation of the list of decision rules is explained in more details in the algorithm presented below. The algorithm is an adaptation of the CN2 induction algorithm presented in [11].

Algorithm: CN2 Induction Algorithm

(1) Let D be the training set
(2) Let Rule_List=empty be the list of decision rules
(3) **repeat**
(4) R=find_best(D)
(5) D'= all examples covered by R
(6) remove D' from D
(7) let C be the most common class in D'
(8) add $R \Rightarrow C$ to Rule_List
(9) **until** (no R found) or (D empty)
(10) **return** Rule_List

Function find_best(d) returns R
(1) best_set $\leftarrow$ { nil }
(2) best_rule $\leftarrow$ nil
(3) A $\leftarrow$ {all attributes in D}
(i.e. in text categorization all the existing words in the training set)
(4) **repeat**
(5) Candidate_set $\leftarrow$ $\{x \wedge y \,\|x \in$ best_set $\wedge$ y $\in$ A$\}$
(6) remove all tuples from Candidate_set that belong
to best_set or are contradictory (e.g. $a = c_1 \wedge a = c_2$)
(7) **foreach** element C_i in Candidate_set
(8) **if** C_i more informative than best_rule
(9) best_rule $\leftarrow C_i$;
(10) remove from Candidate_set those elements
that are informative lower than a user specified threshold
(11) best_set $\leftarrow$ Candidate_set
(12) **until** best_set == nil

(13) **return** best_rule

The above algorithm searches in the training set for the best pattern. Then, it eliminates from the training set the covered cases by this pattern and add to the rule list a correlation between the pattern and the most common class covered by the pattern. It repeats this step until there are no more documents in the training set or no other pattern can be found. The patterns are discovered in the following way. Consider A as being the set of all attributes in D. Construct the candidate set to best rule by doing all the combinations between an element from best_set (rules that candidate to best_rule) and one from A. From the candidate set remove those tuples that are already in best_set or that are contradictory. When the candidate set is built take each element at one and if it is more informative than the existing best_rule, replace the best_rule with this element. When all the elements in candidate set were tested, return the best_rule.

2.2.5 Example-based Classifiers

Example-based classification methods do not build an actual classifier, but rather base their decision upon the training objects similar to the test ones. They are based on analogy, rather than learning. That is why these methods belong to example-based class and they are called lazy learning systems.

The best known classification method belonging to this class is k-NN (k nearest neighbours). When a new object has to be classified using this method, k-NN searches the k training objects most similar to the new one. The closeness of two objects is defined in terms of Euclidean distance. In text categorization, often, the documents are represented in a vector model, the Euclidean distance being computed between each element of the vectors associated to documents (i.e. the weights associated with each term). The Euclidean distance between two points is given by the formula in Equation 2.3.

$$d(X,Y) = \sqrt{\sum (x_i - y_i)^2} \tag{2.3}$$

A decision is made on these k objects. If a large portion of k samples is

classified under c_i class then the new object falls in that class. The building of a k-NN classification system also involves finding the threshold k. This is usually done experimentally by using a validation set, when the number of similar objects to be considered is determined.

2.2.6 Neural Networks

Artificial neural network models have been studied for many years in the hope of achieving human-like performance in several fields like speech and image understanding. The networks are composed of many non-linear computational elements operating in parallel and arranged in patterns reminiscent of biological neural networks. Computational elements or nodes are connected in several layers (input, hidden and output) via weights that are typically adapted during the training phase to achieve high performance. Instead of performing a set of instructions sequentially as in a Von Neumann computer, neural network models explore simultaneously many hypotheses using parallel networks composed of many computational elements connected by links with variable weights. A typical n-layer neural network is described in Figure 2.2.

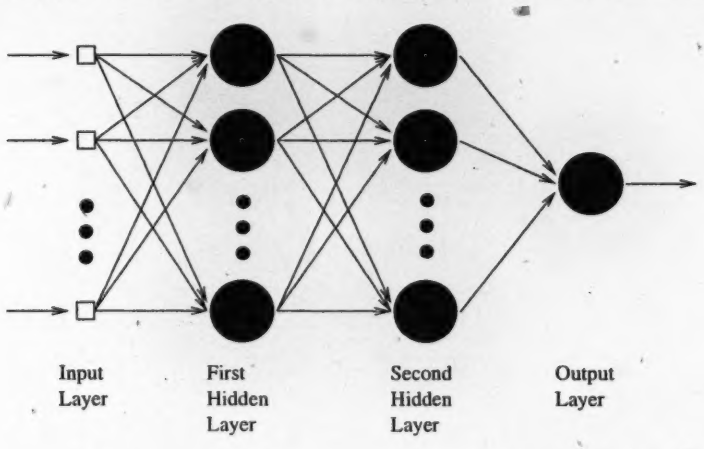


Figure 2.2: N-Layer Neural Network

The weights of a neural network are set during the training phase. The attributes of a training sample are given to the input layer and the weights are automatically adjusted so that the output layer indicates the category assigned to the training object. This process is repeated for all training objects. The

classifier is represented by the neural network with the weights adjusted during the training process. To classify a new object the attributes are assigned to the nodes in the input layer and the value given by the output layer, indicates the class in which the object should fall.

Given a set of training examples to be fed into a neural network the goal is to adjust the weights attached to the nodes in the network in such a manner that makes almost all the tuples in the training data classified correctly.

Below are presented the basic steps in generating a trained neural network.

Training a neural network:

(1) initialize weights with random values
(2) feed the training examples into the network one by one
(3) **foreach** example
(4) compute the input to the network as a linear combination of all the inputs to the input layer
(5) compute the value for the output node using the activation function
(6) evaluate the error between the expected outcome and the predicted value
(7) update the weights and the bias of the network

2.2.7 Support Vector Machines

The concept of support vector machines was introduced in 1995 by Vapnik [50]. This method is based on the *Structural Risk Minimization* principle from computational learning theory. The main idea is to find in the space of data, the hyperplane h that discriminates best between two classes. The samples that lie closest to the hyperplane (both positive and negative examples) are called support vectors. Once the hyperplane is determined, new objects can be classified by checking in which part of the hyperplane they belong to. A graphical representation is given in Figure 2.3.

The problem is to find h with the lowest error. The upper bound of the error is given in Equation 2.4, where n is the number of training examples and d is the Vapnik-Chervonenkis dimension. The VC-dimension characterizes the

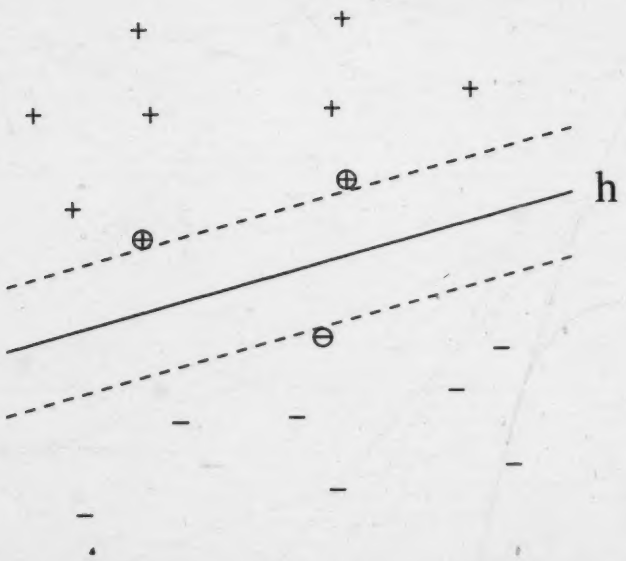


Figure 2.3: Support Vector Machine Classifier

complexity of the problem.

$$P(error(h)) \leq train_error(h) + 2 * \sqrt{\frac{d * (ln\frac{2*n}{d} + 1) - ln\frac{\eta}{4}}{n}} \tag{2.4}$$

The method generates a number of hypothesis H_i so that the VC-dimension increases. The idea is to find the hypothesis that minimizes equation 2.4. When the optimal hyperplane was found for each class, the classification phase is trivial. For each new object to be classified it is checked on which part of the hyperplane it falls, and that category is assigned to it. SVM is a typical approach were the multi-class problem is diveded into disjoint binary categorization tasks. To classifiy a new object all binary classifiers are envoked and their decisions are combined to predict the new classes associated with the object to be classified.

2.2.8 Boosting and Bagging

The technique of combining the prediction of multiple classifiers to build a single classifier was studied by the researchers based on the idea that k classification systems can perform better than a single one. This process is called

voting. Such a classifier takes a learning method and a training set and builds multiple classifiers on different versions of the training set. These resulting classifiers are combined to generate the final one. Voting algorithm can be divided in two classes: boosting and bagging.

Boosting The boosting method [44] builds a classifier by using multiple previously generated classifiers. By using the same learning method it induces k classifiers. The difference among these systems is the set used as training. It differs from one set to another. The method selects N samples from the training set when creating a new classifier. However, the probability of choosing a sample is not the same for all classifiers. It depends on how often the sample was misclassified with the previous created classifiers. Thus the boosting method attempts to create at each stage better classifiers in order to increase the performances of the final classifier. Many methods employing boosting algorithms were presented in the literature. They differ in the way the samples are selected when a new classifier is to be generated.

Bagging The bagging method [9], as well as boosting, takes as input a learning method and a training set of objects. The difference consists of how the samples are chosen for building the classifiers. The training sample has the same size for each classifier as the original training set. For each classifier, the algorithm makes a number of replacements in the training set by uniform probability random selection. This means that some samples could repeat in the training set, while others may not be present at all.

To classify a new object, all the classifiers built are invoked. The new object falls in the class that obtained the most votes.

2.2.9 The Rocchio Classifier

Rocchio's algorithm [29] is a classical method in information retrieval, being used in routing and filtering documents. It relies upon the Rocchio's formula for relevance feedback in the vector space model.

According to Formula 2.5 a vector is computed for each category $\{w_1, w_2, ... w_n\}$.

$$w_i = (\frac{\beta}{+} * \sum(d_i)) - (\frac{\gamma}{-} * \sum(d_j)) \tag{2.5}$$

In Formula 2.5, β and γ are control parameters that allow to adjust the importance of positive (+) or negative (-) examples. When a new object has to be classified, a similarity measure is computed between the object and the representative vectors for each category. The new object is assigned the label of the class that has the closest vector.

2.2.10 N-grams vs. Unigrams

Some researchers attempted to improve the categorization performance by extracting and using phrases. This idea, as well as the previous explained method, is text-oriented. The main idea is to extract from textual databases high quality phrases to construct a classifier. The use of phrases was extensively studied in the literature [14, 33]. In a number of experiments it was proved that the use of phrases actually decreased the performance. However, in [46] it is claimed that the use of bigrams (two-word phrases) improve the quality of results when used instead of unigrams (one-word phrases).

An overall comparison of the discusses classifiers is given in Table 2.2. We compare them with respect to learning and testing phase, as well as from the feature space dimensionality point of view and from the understandability of the model.

Classification method	Learning phase	Testing phase	Feature space	Understand-ability	Model	Reported applications
Associative	fast	fast	high	good	rules	other
Probabilistic(NB)	moderate	fast	low	difficult	network	other+doc
Decision trees	moderate	fast	low	good	rules	other+doc
Decision rules	moderate	fast	low	good	rules	other+doc
Example-based (kNN)	NA	slow	low	good	similarity	other+doc
Neural Networks	slow	fast	high	difficult	network	other+doc
SVM	moderate	fast	high	difficult	hyperplanes	other+doc
Rocchio	fast	fast	high	difficult	similarity	doc
N-grams	moderate	fast	low	good	rules	doc

Table 2.2: Classification methods comparison

2.3 Document Categorization

2.3.1 Text Categorization

Automatic text categorization has always been an important application and research topic since the inception of digital documents. Text categorization research has a long history, starting in the early 1960s. Nowadays, with all the textual information on the Web or in companies' intranets, text categorization has been revived and there is more demand for effective and efficient classification models.

The text classification task can be defined as assigning category labels to new documents based on the knowledge gained in a classification system at the training stage. In the training phase we are given a set of documents with class labels attached and a classification system is built using a learning method. However, a preprocessing phase is needed before the learning method is applied. Its main goal is to select the representative features that characterize a document.

Most of the research in the text categorization area has been devoted to single class categorization (i.e. a document is classified as relevant to one and only topic). However, most of the existing textual information point to a multi-class categorization (i.e. a document can have many class labels). The most common solution to this problem is to divide the multi-class task into disjoint single-class problems.

(1) a preprocessing phase of the text collection
(2) choosing the document representation (indexing) and dimensionality reduction
(3) applying a learning method on the training set
(4) evaluating the obtained classifier on the testing set

The preprocessing phase is represented usually by stopwords removal (i.e. terms that are too frequent in the text collection and insignificant) and by applying stemming (i.e. reduce the words to their canonical form).

The next phase in building a text categorization system is to choose a representation model. Generally, text categorization systems use a vector model representation of the documents. The vector that represents the document contains the document terms and also the weights assigned to each term. The determination of the weights can be done in several ways: boolean weighting, TF-IDF weighting, entropy weighting, etc. In addition, at this stage, a dimensionality reduction is employed in some approaches. This is done by applying some thresholds based on term frequency, information gain, chi-square test or by employing latent semantic analysis to re-parameterize the feature space.

Different approaches have been proposed in the literature for building text categorization systems.

Most of the research in text categorization comes from the machine learning and information retrieval communities. Classifiers based on probabilistic models have been proposed starting with the first presented in the literature by Maron in 1961 [38] and continuing with Naïve-Bayes [32] that proved to perform well. Rocchio's algorithm [29] is the classical method in information retrieval, being used in routing and filtering documents. ID3 and C4.5 are well-known algorithms whose cores are making use of decision trees to build automatic classifiers [12, 13, 30, 5]. K-nearest neighbour is another technique that was used in text categorization [53]. Another method to construct a text categorization system is by an inductive rule learning method [40, 34, 4, 5]. As reported in [46] the use of bigrams improved text categorization accuracy as opposed to unigrams use. In addition, in the last decade neural networks and support vector machines were used in text categorization and they proved to be powerful tools [43, 54, 30]. Voting algorithms were used in some text categorization systems [24, 23].A thorough survey of text categorizati n systems pres nted in the literature is given in [45].

2.3.2 Image Classification

Image classification is the task of categorizing images in predefined classes using an image classification model. The categorization of images has multiple applications including: organization of digital libraries, image retrieval,

automatic annotations of images, pattern recognition, etc.

A common way of building an image classification model follows a number of steps: feature extraction, organization of the features in a database and using a learning method to develop a classification system from the given data. In the feature extraction step many methods have been employed. (i.e. statistical parameters, entropy-based, histogram, etc.) As learning methods, decision trees, decision rules, probabilistic-based, and neural networks have been applied.

Different approaches have been used for building image categorization systems. In addition, image classification systems have many real-world applications in which they have been tested.

Classifiers based on probabilistic models have been proposed to classify images. In [48] a Bayesian classifier is used to classify images with the goal of image indexing. A probabilistic model (i.e. singular value decomposition) is used in [21] to determine the volcanoes on Venus and to annotate the images as containing volcanoes or not.

In [21] an extension of ID3 and an improved decision rule algorithm are employed to classify sky survey images.

K-nearest neighbour is another technique that was used in image classification. The classification is example-based, new images being classified under a certain class based on how close it was to the examples belonging to that class.

In addition, neural networks and support vector machines proved to be powerful tools. Many systems based on neural networks have been used in medical imaging [19, 15].

Chapter 3

Association Rule-based Classification

This chapter describes in detail the classification method that we propose. In our algorithm, as we shall see in the following sections, we take advantage of the apriori algorithm (presented in Section 2.1.2) to discover frequent itemsets in a transactional database . Eventually, these frequent itemsets associated with categories represent the discriminate features among the objects in the collection. The association rules discovered in this stage of the process are further processed to build the associative classifier. We propose two algorithms to generate the association rules that will be further processed to build the categorization system. More details are given in Section 3.1.

Given a set of training documents a transactional database is constructed as follows. Consider a document D_i in the training set that contains the set of words $W = \{w_1, w_2, ...w_n\}$. To this document a set of categories $C = \{c_1, c_2, ...c_m\}$ is attached. This document is represented in the database as the following transaction: $D_i : \{c_1, c_2, ...c_m, w_1, w_2, ...w_n\}$.

After the preprocessing phase, only a set of representative features $F = \{f_1, f_2, ...f_n\}$ is retained, then the transaction is remodelled as the object: $O_i : \{c_1, c_2, ...c_m, f_1, f_2, ...f_n\}$ and the association rules are discovered from these transactions.

Using the apriori algorithm on our transactions representing the objects in the database would generate a very large number of association rules, most of them irrelevant for classification. We use an apriori-based algorithm that

is guided by the constraints on the rules we want to discover. When association rules are employed for classification, we are interested to discover only association rules that have as antecedent a set of features and as consequent a class label. Moreover, pruning techniques are invoked on the rules generated, so that the classification system consists only of the best representative ones. Pruning techniques are presented in Section 3.2.

3.1 Association Rule Generation

There are two approaches that we have considered in generating rules to build an associative classifier. The first one, Association Rule-based Classifier with All Categories (ARC-AC) is to extract association rules from the entire training set. This algorithm is presented in the next subsection. As a result of discrepancies among the categories in a text collection of a real-world application, we discovered that it is difficult to handle some categories that have different characteristics (small categories, overlapping categories or some categories have objects that are more correlated than others). Therefore we propose a second solution, Association Rule-based Classifier By Category (ARC-BC) that would solve such problems. In this approach we consider each set of documents belonging to one category as a separate collection.

ARC-AC has the advantage that it is applied to the entire training set, while ARC-BC is more advantageous when unbalanced data sets are presented to be classified.

In both algorithms we use a constraint, so that only the rules that could be used for classification are generated. In other words, given the transaction model described above, we are interested in rules of the form $f_x \Rightarrow c_i$ where $f_x \subseteq F$ and $c_i \subseteq C$ where F is the set of features and C is the set of class labels. To discover these interesting rules efficiently we push the rule shape constraint in the candidate generation phase of the *apriori* algorithm in order to retain only the suitable candidate itemsets. Moreover, at the phase for rule generation from all the frequent k-itemsets, we use the rule shape constraint again to weed out those rules that are of no use in our classification.

The algorithms for these two approaches are described in more detail in the next sections.

3.1.1 Association Rule-based Classification with All Categories

This section introduces the rule generation phase of building an associative classifier when the rules are extracted from the entire training set at once. In this approach (Figure 3.1) all the transactions in the database form a single training collection and the rules generated are *de facto* the classifier.

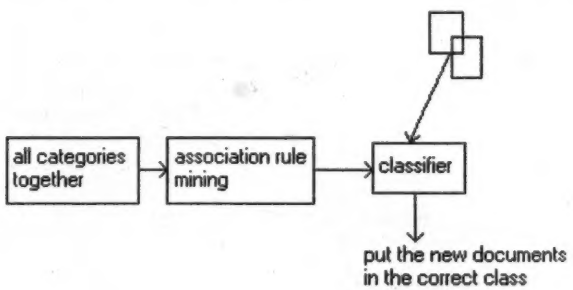


Figure 3.1: Classifier for all categories

Let us consider the following example to illustrate how the ARC-AC algorithm works. In the following example the table containsthe transaction ID in the first column, the class label in the second and the attributes of each transaction in the third column.

Example 3.1 ARC-AC Let consider the following transaction table:

Trans ID	Class Label	Attributes
1	C1	A B
2	C1	A C
3	C2	B D E
4	C2	B C E
5	C2	B C D E
6	C2	B D
7	C2	C D E
8	C3	C F

Table 3.1: A transactional database

The set of 1-itemsets among the attributes in the transaction table (Table 3.1) are presented in the Table 3.2.

1-itemset	support	possible correlations between the 1-itemset and a class label
A	2	$A \Rightarrow C1$
B	5	$B \Rightarrow C1$
		$B \Rightarrow C2$
C	5	$C \Rightarrow C1$
		$C \Rightarrow C2$
		$C \Rightarrow C3$
D	4	$D \Rightarrow C2$
E	4	$E \Rightarrow C2$
F	1	$F \Rightarrow C3$

Table 3.2: 1-itemsets from the transaction table 3.1 with their supports and possible correlations with the class labels

Let us consider the minimum support 25% (2 out of 8) and minimum confidence 50%. In this case, F:1 1-itemset is eliminated. Only one association rule that has as consequent C3 category is remaining. However, this rule ($C \Rightarrow C3$ confidence 20%) is not a strong one; it doesn't exceed the minimum confidence threshold. As a result, with the given support and confidence thresholds there are no rules generated for C3 category. Let us consider the minimum support 50% (4 out of 8) and minimum confidence 50%. In this case, A:2, F:1 1-itemsets are eliminated. This is the set of rules that point to C1 and C3 categories: $B \Rightarrow C1$ confidence 20%, $C \Rightarrow C1$ confidence 20%, $C \Rightarrow C3$ confidence 20%. None of these rules passes the minimum confidence threshold. In conclusion, there will be no rules generated for C1 and C3 classes. This is the result using ARC-AC algorithm. The lack of rules for some small categories can be overcome by employing ARC-BC algorithm that will be discussed in the next section.

The following algorithm presents step by step the process of discovering association rules when the training set is mined at once.

Algorithm ARC-AC Find association rules on the training set of the data collection

Input A set of objects $\mathcal{O}_1$ of the form $\mathcal{O}_i : \{cat_1, cat_2, ...cat_m, f_1, f_2, ...f_n\}$

where cat_i is a category attached to the object and f_j are the selected features for the object; A minimum support threshold σ;

Output A set of association rules of the form $f_1 \wedge f_2 \wedge ... \wedge f_n \Rightarrow cat_i$ where cat_i is a category and f_j is a feature;

Method:

(1) $C_0 \leftarrow$ {Candidate categories and their *support*}
(2) $F_0 \leftarrow$ {*Frequent* categories and their *support*}
(3) $C_1 \leftarrow$ {Candidate 1 itemsets and their *support*}
(4) $F_1 \leftarrow$ {*Frequent* 1 itemsets and their *support*}
(5) $C_2 \leftarrow$ { candidate pairs (cat, f) such that (cat, f) $\in \mathcal{O}_1$ and $cat \in F_0$ and $f \in F_1$}
(6) foreach object o in $\mathcal{O}_1$ do {
(7) foreach $c = (cat, f)$ in C_2 do {
(8) $c.support \leftarrow c.support + Count(c, o)$
(9) }
(10) }
(11) $F_2 \leftarrow \{c \in C_2 \mid c.support > \sigma\}$
(12) $\mathcal{O}_2 \leftarrow FilterTable(\mathcal{O}_1, F_2)$
(13) for $(i \leftarrow 3; F_{i-1} \neq \emptyset; i \leftarrow i+1)$ do {
(14) $C_i \leftarrow (F_{i-1} \bowtie F_2)$ /* $\forall c \in C_i$ c has only one category */
(15) $C_i \leftarrow C_i - \{c \mid (i-1)$ item-set of $c \notin F_{i-1}\}$
(16) $\mathcal{O}_i \leftarrow FilterTable(\mathcal{O}_{i-1}, F_{i-1})$
(17) foreach object o in $\mathcal{O}_i$ do {
(18) foreach c in C_i do {
(19) $c.support \leftarrow c.support + Count(c, o)$
(20) }
(21) }
(22) $F_i \leftarrow \{c \in C_i \mid c.support > \sigma\}$
(23) }
(24) Sets $\leftarrow \bigcup_i \{c \in F_i \mid i > 1\}$
(25) R= $\emptyset$

(26) foreach itemset I in Sets do {

(27) $\quad R \leftarrow R + \{f \Rightarrow c \mid f \cup c \in I \wedge f$ is an itemset $\wedge c \in C_0\}$

(28) }

Algorithm ARC-AC generates the strong rules when the entire collection is mined. In steps (1-10) the two-frequent itemsets are generated by joining the frequent categories and frequent 1-itemset. Step (11) retains only those that exceed the minimum support threshold. In (13-23) all the k-frequent itemsets are discovered as explained in the apriori algorithm. The last 4 steps represent the actual association rule generation stage. With both algorithms, ARC-AC and ARC-BC, the document space is reduced in each iteration by eliminating the transactions that do not contain any of the frequent itemsets. This step is done by *FilterTable*($\mathcal{O}_{i-1}$, F_{i-1}) function.

3.1.2 Association Rule-based Classification by Category

The second model that we propose takes advantage of the knowledge given in the training set. In the training database to each tuple is associated a set of classes in which that tuple may fall. This is the knowledge that we take advantage of. We use this knowledge to discover association rules that have as antecedent a set of features and as consequence a class label. By building this model we try to find a high quality set of rules for each category. In this approach (Figure 3.2), each class is considered as a separate training collection and the association rule mining applied to it. In this case, the transactions that model the training documents are simplified to $O_i : \{C, t_1, t_2, ...t_n\}$ where C is the category considered.

Let us consider again the transactional database from Example 3.1. Having as class labels three distinct categories leads to three training sets when ARC-BC is employed. Each training set consists of tuples belonging to the same category. The next tables show the training sets when the transactional database from Example 3.1 is used. As is presented below, by applying the association rule mining phase by category does not eliminate the attributes

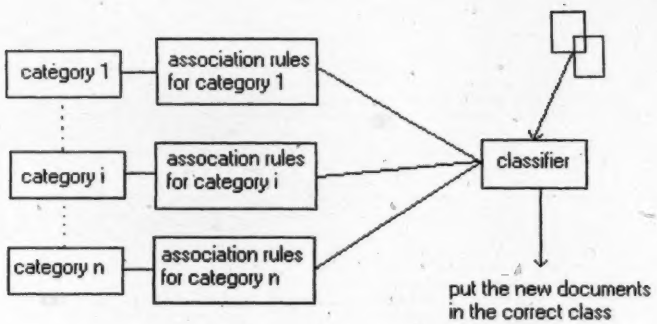


Figure 3.2: Classifier per category

that are specific for a certain category. They have, however, a low support with respect to the entire collection.

Table 3.3: The Training set for C1

Trans ID	Class Label	Attributes
1	C1	A B
2	C1	A C

Mining the tuples in Table 3.3 we have the following set of 1-itemsets: A:2, B:1, C:1 and the following set of rules is generated:

(1) $A \Rightarrow C1$ conf=2/2=100%

(2) $B \Rightarrow C1$ conf=1/1=100%

(3) $C \Rightarrow C1$ conf=1/1=100%

Table 3.4: The Training set for C2

Trans ID	Class Label	Attributes
3	C2	B D E
4	C2	B C E
5	C2	B C D E
6	C2	B D
7	C2	C D E

Mining the tuples in Table 3.4 we have the following set of 1-itemsets: B:4, C:3, D:4, E:4 and the following set of rules is generated:

(1) $B \Rightarrow C2$ conf=4/5=80%

(2) $C \Rightarrow C2$ conf=3/5=60%

(3) $D \Rightarrow C2$ conf=4/4=100%

(4) $E \Rightarrow C2$ conf=4/4=100%

Table 3.5: The Training set for C3

Trans ID	Class Label	Attributes
8	C3	C F

Mining the tuples in Table 3.5 we have the following set of 1-itemsets: C:1, F:1 and the following set of rules is generated:

(1) $C \Rightarrow C3$ conf=1/1=100%

(2) $F \Rightarrow C3$ conf=1/1=100%

In both cases, when the minimum support was set to 25% or 50% and minimum confidence to 50%, there have been generated rules for all three categories.

Based on the observations presented above we developed the next algorithm.

Algorithm ARC-BC Find association rules on the training set of the transactional database when the collection is divided in subsets by category

Input A set of objects $\mathcal{O}$ of the form $\mathcal{O}_1 : \{cat, f_1, f_2, ...f_n\}$ where *cat* is the category attached to the subset and f_j are the selected features for the object; A minimum support threshold σ;

Output A set of association rules of the form $f_1 \wedge f_2 \wedge ... \wedge f_n \Rightarrow cat$ where *cat* is the category and f_j is a feature;

Method:

(1) $C_1 \leftarrow \{$Candidate 1 itemsets and their *support*$\}$

(2) $F_1 \leftarrow \{Frequent$ 1 itemsets and their *support*$\}$

(3) for $(i \leftarrow 2; F_{i-1} \neq \emptyset; i \leftarrow i+1)$ *do*$\{$

(4) $\quad C_i \leftarrow (F_{i-1} \bowtie F_{i-1})$

(5) $\quad C_i \leftarrow C_i - \{c \mid (i-1)$ item-set of $c \notin F_{i-1}\}$

(6) $\quad \mathcal{O}_i \leftarrow FilterTable(\mathcal{O}_{i-1}, F_{i-1})$

(7) $\quad$ foreach object o in $\mathcal{O}_i$ do $\{$

(8) foreach c in $\dot{C}_i$ do {

(9) $c.support \leftarrow c.support + Count(c,\ o)$

(10) }

(11) }

(12) $F_i \leftarrow \{c \in C_i \mid c.support > \sigma\}$

(13) }

(14) Sets $\leftarrow \bigcup_i \{c \in F_i \mid i > 1\}$

(15) R= $\emptyset$

(16) foreach itemset I in Sets do {

(17) $R \leftarrow R + \{I \Rightarrow cat\}$

(18) }

In ARC-BC algorithm step (2) generates the frequent 1-itemset. In steps (3-13) all the k-frequent itemsets are generated and the category *cat* is added at the end. Steps (16-18) generate the association rules.

3.2 Pruning the Set of Association Rules

The number of rules that can be generated in the association rule mining phase could be very large. There are two issues that must be addressed in this case. One of them is that such a huge amount of rules could contain noisy information which would mislead the classification process. Another is that a huge set of rules would make the classification time longer. This could be an important problem in applications where fast responses are required.

The pruning methods that we employ in this project are the following: eliminate the specific rules and keep only those that are general and with high confidence, and prune unnecessary rules by database coverage. More details are given in the following subsections.

3.2.1 Removing Low Ranked Specialized Rules

Removing low ranked specialized rules is one of the pruning techniques employed in our research. The use of this pruning techniques is motivated by

the fact that if we keep general rules with high confidence it is enough for the classification process. Let's take the following example: $R_1 : F_1 \Rightarrow C$ having 90% confidence and $R_2 : F_1 \wedge F_2 \Rightarrow C$ with 80% confidence. In this case R_1 will cover all the examples that will be covered by R_2, having in common the feature F_1. However, R_1 has a higher confidence which makes the relationship between the feature and the class even stronger. That is why pruning R_2 would not affect the classification process.

Let us introduce the notions used in this subsection by the following definitions:

Definition1 Given two rules $R_1 \Rightarrow C$ and $R_2 \Rightarrow C$ we say that the first rule is more general if $R_1 \subseteq R_2$.

The first step of this process is to order the set of rules. This is done accordingly to the following ordering definition.

Definition2 Given two rules R_1 and R_2, R_1 is higher ranked than R_2 if:

(1) R_1 has higher confidence than R_2

(2) if the confidences are equal supp(R_1) must exceed supp(R_2)

(3) both confidences and support are equal but R_1 has fewer attributes in left hand side than R_2

The two definitions presented above are similar to those used in CMAR algorithm [35]. With the set of association rules sorted, the goal is to select a subset that will build an efficient and effective classifier. In our approach we attempt to select a high quality subset of rules by selecting those rules that are general and have high confidence.

3.2.2 Pruning Rules based on Database Coverage

The most significant subset of rules is finally selected by applying the database coverage. The idea is to select a high quality set of rules that can cover the entire training set of data. Each tuple in the training set should be covered by at least one rule in the coverage set. Once the whole data set is covered, those rules that do not belong to the coverage set are pruned. The algorithm for building this set of rules is described below. The algorithm does both the removing of low ranked specialized rules (discussed in the previous section)and

the database coverage.

Algorithm Pruning the set of association rules

Input The set of association rules that were found in the association rule mining phase (S) and the training text collection (D)

Output A set of rules used in the classification process

Method:

(1) sort the rules according to **Definition1**
(2) **foreach** rule in the set S
(3) find all those rules that are more specific
(4) prune those that have lower confidence
(5) a new set of rules S' is generated
(6) **foreach** rule R in the set S'
(7) go over D and find those transactions that are covered by the rule R
(8) **if** R classifies correctly at least one transaction
(9) select R
(10) remove those cases that were covered by R

In line 1, the rules are sorted according to Definition1. In lines 2-4 the low ranked specialized rules are removed. Starting at line 6 database coverage is employed (see Section 2.2.1 for database coverage algorithm). At the end of this algorithm a high quality set of rules is selected which will represent the classifier.

3.2.3 Other pruning techniques

There is another pruning technique whose effect we studied in this work. This pruning method employed is to eliminate conflicting rules, rules that for the same characteristics would point to different categories. For example, given two rules $T_1 \Rightarrow C_1$ and $T_1 \Rightarrow C_2$ we say that these are conflicting since they could introduce errors. This technique reduces the number of rules discovered at the mining stage. In addition, it can affect the accuracy result of the classifier.

3.3 Prediction of Classes Associated with New Objects

The set of rules that were selected after the pruning phase represent the actual classifier. This categorizer will be used to predict to which classes new documents are attached. Given a new document, the classification process searches in this set of rules for finding those classes that are relevant to be attached with the document presented for categorization. This subsection discusses the approach for labelling new documents based on the set of association rules that forms the classifier.

A trivial solution would be to attach to the new document the class that has the most rules matching this new document or the class associated with the first rule that apply to the new object.

Let us consider the following example.

Example: S is a set of rules that represents the classifier and O is a new object to be classified. Let object O have the following features $< f_1, f_3, f_4, f_7, f_9 >$. From the set of rules S, the following rules cover this tuple:

$f_1 \Rightarrow C_1$ confidence 0.9

$f_3 \wedge f_4 \Rightarrow C_2$ confidence 0.85

$f_4 \Rightarrow C_2$ confidence 0.8

$f_7 \Rightarrow C_1$ confidence 0.6

$f_9 \Rightarrow C_3$ confidence 0.5

In this example, if we classify D according to the first ranked rule applied will be classified under C_1. When the most number of rules is taken into consideration a decision must be made between C_1 and C_2 classes. However, the second and third rule point to C_2 and together they have high confidence. This observation may lead to the question "is it the decision reliable or not?" or "how about multi-class categorization?".

However, in many domains, text categorization being one of them, multi-class categorization is an important and challenging problem that needs to be solved. In our approach we give a solution to this problem by introducing the *dominance factor*. By employing this variable we allow our system to assign

more than one category. The dominance factor δ is the proportion of rules of the most dominant category in the applicable rules for a document to classify. Given a document to classify, the terms in the document would yield a list of applicable rules. If the applicable rules are grouped by category in their consequent part and the groups are ordered by the (sum of rules' confidences divided by the number of rules, the ordered groups would indicate the most significant categories that should be attached to the document to be classified. We call this order category dominance, hence the dominance factor δ. The dominance factor allows us to select among the candidate categories only the most significant. When δ is set to a certain percentage, only the categories that have enough applicable rules representing that percentage of the number of rules applicable for the most dominant category are selected.

Let us consider again the above example. The following groups are found when the applicable rules are grouped by category in their consequent part.

$f_1 \Rightarrow C_1$ confidence 0.9

$f_7 \Rightarrow C_1$ confidence 0.6

$f_3 \wedge f_4 \Rightarrow C_2$ confidence 0.85

$f_4 \Rightarrow C_2$ confidence 0.8

$f_9 \Rightarrow C_3$ confidence 0.5

When the rules are ordered by their sum of confidences divided by the number of rules, the list obtained is the following:

C_2 0.825

C_1 0.75

C_3 0.5

C_2 is the dominant class in this case. Using the *dominance factor* we set the threshold above which the classes are considered significant and the predicted class can be made. The *dominance factor* allows us to perform both single-class and multiple-class categorization. If we want to have single-class classification the dominance factor is set to 100%, in which case the threshold is set to the value of the dominant class. In this case, for the example above the new tuple O is predicted to be in class C_2 (threshold take the value 0.825 * 100/100). Otherwise, if the task is multiple-class categorization the dominance

factor is set to an important threshold. Let us consider the *dominance factor* at 80% (now the threshold is set to 0.825*80/100=0.6). In this case, the object O is classified under C_2 and C_1, since C_1 has a combined confidence 0.75 higher than 0.6 while the combined confidence of C_3 is 0.5 which is lower than 0.6.

The next algorithm describes the classification of a new document. TakeKClasses(S,δ) function in the algorithm below selects the most k significant classes in the classification algorithm.

Algorithm Classification of a new object

Input A new object to be classified o; The associative classifier (ARC); The dominance factor δ; The confidence threshold τ;

Output Categories attached to the new object

Method:

(1) $S \leftarrow \emptyset$ /*set of rules that match o*/
(2) **foreach** rule r in ARC (the sorted set of rules)
(3) **if** ($r \subset o$) { count++ }
(4) **if** (count == 1)
(5) fr.conf $\leftarrow$ r.conf /*keep the first rule confidence*/
(6) $S \leftarrow S \cup r$
(7) **else if** (r.conf > fr.conf-τ)
(8) $S \leftarrow S \cup r$
(9) **else** exit
(10) divide S in subsets by category: $S_1, S_2 ... S_n$
(11) **foreach** subset $S_1, S_2 ... S_n$
(12) sum the confidences of rules and divide by the number of rules in S_k
(13) **if** it is single class classification
(14) put the new document in the class that has
the highest confidence sum
(15) **else** /*multi-class classification*/
(16) TakeKClasses(S,δ)
(17) assign these k classes to the new document

In the above algorithm, a set of aplicable rules is selected in the lines 1-8. The prediction process is starting at line 9. The applicable set of rules is divided according to the classes in line 10. In lines 11-12 the groups are ordered according to the sum of confidences divided by the number of rules. Starting at line 13 the classification is made. If it is performed a single-class classification the new document is associated to the class that has the highest S_k (see lines 13-14). Otherwise, when a multi-class categorization is employed the new document is assigned to k classes as discused above, using the dominance factor.

3.4 Comparison between our approaches and the other existing associative classifiers

This section gives a comparison between our approaches and the two associative classifiers presented in Section 2.2., in order to emphasize the contributions made in this thesis. All existing methods assume that classes are uniformly represented in the training set while our ARC-BC method makes no assumptions with respect to the distribution in the training set whether the class representation is even or uneven (i.e. with rare classes). Moreover, only our methods can handle the case of multiple-class classification. Both previous methods consider only single-class classification where each object is labeled with one and only one class label.

The first associative classifier presented in the literature was CBA (Classification Based on Associations) in 1998 [37]. In parallel with our work, CMAR (Classification based on Multiple Association Rules) was developed and published in 2001 [35]. We presented in this thesis two new approaches, ARC-AC (Association Rule-based Classification with All Categories) and ARC-BC (Association Rule-based Classification By Category), both based on association rule mining.

As explained in Section 2.2.1, the main steps in building an associative classifier when a training set is given are the following:

1. *Generating the set of association rules from the training set*: In this

phase association rules of the form $setoffeatures \Rightarrow classlabel$ are discovered by using a mining algorithm.

2. *Pruning the set of discovered rules*: The rule generation phase produces a large set of association rules especially when low support is given. Reducing the number of rules is paramount to improve efficiency and even effectiveness. The pruning techniques are a challenging task to discover the best set of rules that can cover the training set. This phase is employed to weed out those rules that may introduce errors or are overfitting in the classification stage.

3. *Classification phase*: This phase makes use of the set of rules from the previous phase to give a good class prediction for new objects.

Based on these steps we give a comparison in Table 3.6 of each stage of the presented methods.

In the rule generation phase CBA mines the training set in an apriori-like fashion, while CMAR uses FP-tree as association rule discovery algorithm. In our first approach, ARC-AC we discover the frequent patters and generate the association rules in a similar manner with CBA. Essentially, altough the mining methods are slightly different, all the three approaches discused so far generate the same association rules for classification. However, in our second approach both the method and the outcome of the mining algorithm is different. This is due to the fact that the mining is done on the subsets of the training set (based on category label), rather than applying an association rule discovery algorithm on the entire training set. In other words, while CBA, CMAR and ARC-AC produce exactly the same original association rules, ARC-BC generates a completly different set of rules.

The next stage in building an associative classifier is to take advantage of some pruning techniques. CBA uses database coverage to prune overfitting rules. CMAR adds to database coverage the removal of low ranked specialized rules. In our approaches we use the same pruning techniques as CMAR, but we also eliminate conflicting rules. The ranking step of the association

rules discovered is done in the same way by all algorithms, ranking the rules according to their confidence and support.

In the classification stage the classification model built in the learning stage is used to categorize new incoming objects. There are two possible types of classification: single-class classification (i.e. only one class is assigned to the new object) and multiple-class classification (i.e. the object to be classified can fall in multiple classes). Both CBA and CMAR perform only single-class classification. CBA puts the new object in the class attached to the first matching rule from the sorted list of association rules. CMAR analyze the set of rules that match the new object. It computes a weighted chi-square and it assigns the category that has the highest value in this analysis [35]. In both our approaches we use a different method: we analyze the set of matching rules to the new object and decide the winner by computing the average of the confidences for each category. In single-class classification the class with the highest value is chosen, while for multi-class classification we introduce the notion of dominance factor. In multi-class classification we put the new object in those classes that have the average of confidences higher than the (highest_value $\times$ dominace_factor). CBA and CMAR were only experimented on uniform datasets. In addition to those, we also performed successful experiments on datasets with rare classes (i.e. classes with very few representatives in the training set).

	CBA	CMAR	ARC-AC	ARC-BC
Year of publication	1998	2001	2001	2002
Rule generation phase	Apriori-like on the entire set	FP-tree on the entire set	Apriori-like on the entire set	Apriori-like **on subsets** (by category)
Pruning phase	Database coverage	Database coverage and removal of low ranked specialized rules	Database coverage, removal of low ranked specialized rules and **elimination of conflicting rules**	Database coverage, removal of low ranked specialized rules and **elimination of conflicting rules**
Ranking of rules	Rank the rules according to their confidence and support	Rank the rules according to their confidence and support	Rank the rules according to their confidence and support	Rank the rules according to their confidence and support
Single-class classification	Apply only the first raked rule	Analyze a set of rules and decide the winner using a weighted chi-square	Analyze a set of rules and decide the winner **using the average of the confidences**	Analyze a set of rules and decide the winner **using the average of the confidences**
Multi-class classification	N/A	N/A	Use the **dominance factor** to allow multiple classes as prediction	Use the **dominance factor** to allow multiple classes as prediction
Reported experiments	UCI datasets (homogenous datasets)	UCI datasets (homogenous datasets)	UCI datasets (homogenous datasets), **medical images and documents (datasets containing rare classes)**	**Medical images and documents (datasets containing rare classes)**

Table 3.6: Comparison among the existing associative classifiers and the proposed methods in this thesis

Chapter 4

Experimental Results: single label classification

This chapter introduces the experimental results when single class classification is performed using ARC algorithms. The algorithms are applied on three types of data collection. Firstly, we test our algorithms on the UCI [7] datasets, secondly, on a medical image database [39] and thirdly on a gene database. Single class classification means that one tuple in the database could fall only in one class.

The motivation for testing our algorithm on UCI datasets is for comparison reasons. This is one of the collections that was used by the other researchers for testing their associative classification algorithms [35, 37]. In addition, by testing our algorithm on a mammographic database, we tried to prove the efficiency of our algorithm when employed on a image collection. Gene categorization is another important task nowadays. It is a challenging problem that many researchers try to solve. We tested one of our algorithms on this task.

4.1 UCI Datasets

We tested our algorithm on some datasets from UCI ML Repository [7]. On each dataset we performed C4.5's shuffle utility [41] for shuffling the datasets. A 10-fold cross validation was performed on each dataset and the results are given as average of the accuracies obtained for each fold. In addition, to have a

fair comparison with the other algorithms that we wanted to compare, we used the same discretization method for continuous attributes as in [37, 35]. The parameters for C4.5 were set to their default values. For all three association rule based methods the minimum support was set to 1% and the minimum confidence to 50%. In our approach the confidence threshold was set at 10% and the dominance factor to 100% (single-label classification performed).

Dataset	No of Attr.	No of Classes	No of Records	ARC-AC(1)	ARC-AC(2)	C4.5	CBA	CMAR
Breast	10	2	699	96.9	96.8	95	96.3	96.4
Crx	15	2	690	87.83	85.94	84.9	84.7	84.9
Diabetes	8	2	768	78.53	77.5	74.2	74.5	75.8
Glass	9	7	214	68.28	66.91	68.7	73.9	70.1
Heart	13	2	270	81.4	79.54	80.8	81.9	82.2
Iris	4	3	150	94.37	93.8	95.3	94.7	94
Led7	7	10	3200	73.02	71.9	73.5	71.9	72.5
Pima	8	2	768	78.27	78.1	75.5	72.9	75.1
Avg	9.25	3.75	844.875	82.32	81.31	80.98	81.35	81.37

Table 4.1: The success rate comparison of ARC-AC, C4.5, CBA and CMAR

The difference between the two columns for ARC-AC is as follows: in the first column (ARC-AC(1)) the set of rules that form the classifier is the set of rules extracted at the mining stage but ordered according to the confidence and support of the rules (support was normalized so that the ordering is possible even if the association rules are found by category)(see Table 4.1 columns under 'ARC-AC(1)'); in the next column (see Table 4.1 columns under 'ARC-AC(2)') from the ordered set of rules the specific ones were removed if they had lower confidence (see Section 3.2.1). As observed in Table 4.1 we outperform the other two associative classifiers and the state-of-the-art C4.5. We reported our results on some of the UCI datasets for comparison reasons with CBA and CMAR. Due to the small size of UCI datasets and balanced data among classes,we did not employ the ARC-BC algorithm on this datasets.However, our main goal in this thesis is not only to present a new associative classifier, but also to show our contribution in medical image and text classification. More detailed results on these two tasks are given in the next section and in

the next chapter.

4.2 Mammogram Collection

In the following sections we present how our algorithm performed when applied on a mammogram collection. Figure 4.1 shows an overview of the of the classification process when association rule mining is applied on an image dataset.

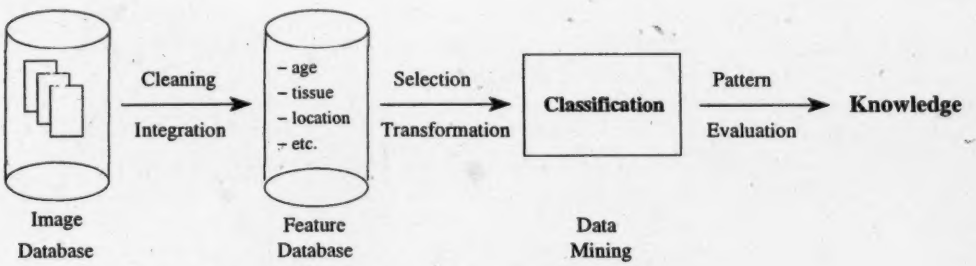


Figure 4.1: Image categorization process

The high incidence of breast cancer in women, especially from developed countries, has increased significantly in recent years. The etiologies of this disease are not clear and neither are the reasons for the increased number of cases. Currently there are no methods to prevent breast cancer, that is why early detection represents a very important factor in cancer treatment and allows reaching a high survival rate. Mammograms are considered the most reliable method in early detection of cancer. Due to the high volume of mammograms to be read by physicians, the accuracy rate tends to decrease and automatic reading of digital mammograms becomes highly desirable. It has been proven that double reading of mammograms (consecutive reading by two physicians or radiologists) increased the accuracy, but at high costs. That is why the computer aided diagnosis systems are necessary to assist the medical staff to achieve high efficiency and effectiveness.

Important visual clues of breast cancer include preliminary signs of masses and calcification clusters. Unfortunately, at the early stages of breast cancer, these signs are very subtle and varied in appearance, making diagnosis difficult,

challenging even for specialists. This is the main reason for the development of classification systems to assist specialists in medical institutions. Since the data that physicians and radiologists must deal with increased significantly, there has been a great deal of research done in the field of medical images classification. With all this effort, there is still no widely used method to classify medical images. This is because this domain requires high accuracy. Also misclassifications could have different consequences. False negatives could lead to death while false positives have a high cost and could cause detrimental effects on patients. For automatic medical image classification, the rate of false negatives has to be very low if not zero. It is important to mention that manual classification of medical images by professionals is also prone to errors and the accuracy is far from perfect.

In addition, the existing tumors are of different types. These tumors are of different shapes and some of them have the characteristics of normal tissue. All these things make the decisions that are made on such images even more difficult. Different methods have been used to classify and detect anomalies in medical images, such as wavelets [10, 51], fractal theory [18], statistical methods [16] and most of them used features extracted using image processing techniques [31]. In addition, some other methods were presented in the literature based on fuzzy set theory [8], Markov models [17] and neural networks [15, 20]. Most of the computer-aided methods proved to be powerful tools that could assist medical staff in hospitals and lead to better results in diagnosing a patient.

4.2.1 Database Description

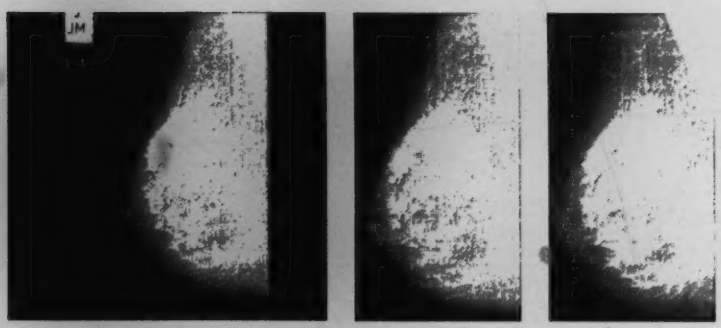
To have access to real medical images for experimentation is a very difficult undertaking due to privacy issues and heavy bureaucratic hurdles. The data collection used in our experiments was taken from the Mammographic Image Analysis Society (MIAS) [39]. Its corpus consists of 322 images, which belong to three big categories: normal, benign and malignant. There are 208 normal images, 63 benign and 51 malign, which are considered abnormal. In addition, the abnormal cases are further divided in six categories: microcalcification,

circumscribed masses, spiculated masses, ill-defined masses, architectural distortion and asymmetry. All the images also include the locations of any abnormalities that may be present. The existing data in the collection consists of the location of the abnormality (like the center of a circle surrounding the tumor), its radius, breast position (left or right), type of breast tissues (fatty, fatty-glandular and dense) and tumor type if exists (benign or malignant). All the mammograms are medio-lateral oblique view.

4.2.2 Preprocessing Techniques

It is well known that data has to be pre-processed before applying any of the Data Mining techniques to improve the effectiveness of the results obtained from mining. Since real-life data is often incomplete, noisy and inconsistent, pre-processing becomes a necessity [28]. Two pre-processing techniques, namely Data Cleaning and Data Transformation, were applied to the image collection. Data Cleaning is the process of cleaning the data by removing noise, outliers etc. that could mislead the actual mining process. In our case, we had images that were very large (typical size was 1024 x 1024) and almost 50% of the whole image comprised of the background with a lot of noise. In addition, these images were scanned at different illumination conditions, and therefore some images appeared too bright and some were too dark. The first step towards noise removal was pruning the images with the help of the crop operation in Image Processing. Cropping cuts off the unwanted portions of the image. Thus, we eliminated almost all the background information and most of the noise. An example of cropping that eliminates the artefacts and the black background is given in Figure 4.2.

Since the resultant images had different sizes, the x and the y coordinates were normalized to a value between 0 and 255. The cropping operation was done automatically by sweeping horizontally through the image. The next step towards pre-processing the images was using image enhancement techniques. Image enhancement helps in qualitative improvement of the image with respect to a specific application [26]. Enhancement can be done either in the spatial domain or in the frequency domain. Here we work with the spatial domain



(a) (b) (c)

Figure 4.2: Pre-processing phase on an example image: (a) original image; (b) crop operation; (c) histogram equalisation

and directly deal with the image plane itself. In order to diminish the effect of over-brightness or over-darkness in images, and at the same time accentuate the image features, we applied the Histogram Equalization method, which is a widely used technique. The noise removal step was necessary before this enhancement because, otherwise, it would also result in enhancement of noise. Histogram Equalization increases the contrast range in an image by increasing the dynamic range of grey levels [26]. Figure 4.2 shows an example of histogram equalisation after cropping.

Feature Extraction

Feature extraction phase is needed in order to create the transactional database to be mined. The features that were extracted were organized in a database, which is the input for both classification systems used. The extracted features are: four statistical parameters: mean, variance, skewness and kurtosis. These features are by no means the best features for mammography analysis or the discrimination of images in general. We use these features only for the sake of comparison studies with other previously published mammography classification techniques.

The general formula for the statistical parameters computed is the following:

$$M_n = \frac{\sum (x - \overline{x})^n}{N} \tag{4.1}$$

where N is the number of data points, and n is the order of the moment. The skewness can be defined as:

$$Sk = \frac{1}{N} * \left(\frac{(x - \overline{x})}{\sigma} \right)^3 \tag{4.2}$$

and the kurtosis as:

$$kurt = \frac{1}{N} * \left(\frac{(x - \overline{x})}{\sigma} \right)^4 - 3 \tag{4.3}$$

All the extracted features presented above have been computed over smaller windows of the original image. The original image was split initially in four parts, as shown in Figure 4.3, for a better localization of the region of interest. In addition, the features extracted were discretized over intervals before organizing the transactional data set.

Figure 4.3: Mammography division

4.2.3 Transactional Database Organization for ARC-AC

When using ARC-AC the features of all quadrants were kept regardless of whether they were normal or cancerous [3]. In addition some other descriptors from the original database were attached, such as breast position, type of tissue, etc.

This database was constructed by merging some already existing features in the original database with some new features that we extracted from the

medical images using image-processing techniques. The existing features were: The type of the tissue (dense, fatty and fatty-glandular); The position of the breast: left or right. The type of tissue is an important feature to be added to the feature database, being well known the fact that for some types of tissue the recognition is more difficult than for others. The motivation for using these existing features was that by incorporating them the accuracy rate could increase. The extracted features were: Four statistical parameters: mean, variance, skewness and kurtosis.

When all the features were extracted the transactional database to be mined was built in the following way. For each image, all the features extracted were attached to the corresponding transaction (e.g. for the mammogram presented in Figure 4.3 all the features extracted for each quadrant were attached).

4.2.4 Transactional Database Organization for ARC-BC

When all the features were extracted the transactional database to be mined was built in the following way. For the normal images, all the features extracted were attached to the corresponding transaction, while for those characterizing an abnormal mammogram only the features extracted from abnormal parts were attached. (e.g. for the mammogram presented in Figure 4.3 only the features extracted for the NE quadrant (the arrow in the figure points to the tumor) were attached; if the mammogram would have been a normal one the features extracted for all the splits would have been attached). This new data cleaning stage allows us to find higher quality rules, discriminating better among the categories.

When using ARC-BC, in addition to selecting quadrants with tumors from abnormal mammograms, we also dropped those additional features from the database because some of them may not be available in other datasets, while others (breast position) proved to mislead the classification process.

4.2.5 Experimental Results

In our experiments, we considered the 322 images from the MIAS database [39] for both classification systems. From these set of images we considered 90 percent for training the systems and 10 percent for testing them. We considered ten splits of the data collection and computed the results for all of them in order to obtain a more accurate result of the systems potential.

As in any learning process for building a classifier, the classification performed with association rule mining comprised two steps. The first one is represented by the training of the system, while the second one deals with the classification of the new images. In the following subsections are presented the results for mammograms classification when both ARC-AC and ARC-BC are employed.

ARC-AC

In the training phase, the apriori algorithm was applied on the training data and the association rules were extracted. The support was set to 10% and the confidence to 0%. The reason for choosing the 0% percent for the confidence is motivated by the fact that the database has more normal cases (about 70%). The 0% confidence threshold allows us to use the confidence of the rule in the tuning phase of the classifier. In the classification phase, the low and high thresholds of confidence are set such as the maximum recognition rate is reached. The success rate for association rule classifier was 69.11% on average. The results for the ten splits of the database are presented in Table 4.2. For each split we selected about 90% of the dataset for training and the rest for testing. That is 288 images in the training set and 34 images in the testing set. One noticeable advantage of the association rule-based classifier is the time required for training, which is very low compared to other methods such as neural networks.

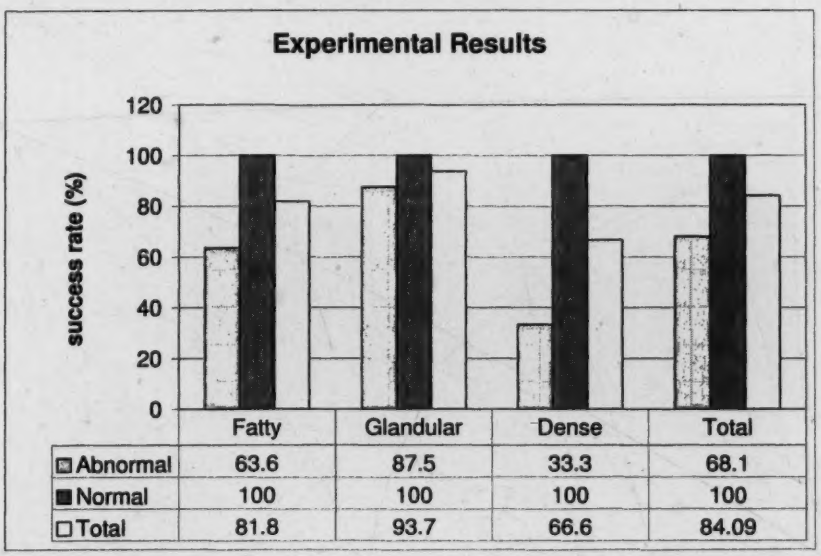
The recognition rate obtained using association rule mining is close to some other results reported in the literature. Another interested fact to be noticed is that the classifier proves to perform well on all the splits of the database, being

Database split	Success ration (percentage)
1	67.647
2	79.412
3	67.647
4	61.765
5	64.706
6	64.706
7	64.706
8	64.706
9	67.647
10	88.235
Average	69.11

Table 4.2: Success ratios for the 10 splits with the association rule based classifier with all categories (ARC-AC)

compact and consistent. The only split that had much higher performance was the 10th one. This was due to the fact that more normal cases belonged to this split than to other. Having a high recognition rate for normal cases the accuracy went high on this split. We noticed that the association rule classifier was sensitive to the unbalanced data collection that contained about 70 percent normal cases and only 30 percent abnormal, this being further divided into benign and malign. This is why we decided to build another classifier using an equilibrated distribution of normal and abnormal cases. For comparison reasons, we used a split that was also chosen in [19]. The same split is not the only reason for choosing [19] as comparison. In addition, the feature extraction phase is similar and a radial basis function network represents the classifier. We considered the 22 mammograms containing circumscribed lesions existing in the database. From these 22 mammograms, there are 18 benign and 4 malign. The abnormal mammograms are further split according to tissue type in fatty (11 cases), fatty-glandular (8 cases) and dense (3 cases). For the training procedure, we have selected 22 abnormal images and 22 normal images selected at random. For the evaluation of the results, we have used all the abnormal mammograms from MIAS database containing circumscribed masses and another 22 normal mammograms randomly selected. For this split the

success rate was better (78.69%) than the previous splits. A noticeable fact is, that due to the imbalance between benign and malign images, the number of rules generated for the malign one was extremely reduced thus all the malign images being misclassified. Three out of four malign images were classified as abnormal (benign) which means that the classification in just normal and abnormal categories was actually higher (84.09%) which is a significant improvement over the results presented in [19] (75.2%). The classification results in normal and abnormal categories are presented in Figure 4.4. As compared to the results presented in [19] we obtained a lower recognition rate for the fatty abnormal mammograms, but higher for all the normal cases.



	Fatty	Glandular	Dense	Total
Abnormal	63.6	87.5	33.3	68.1
Normal	100	100	100	100
Total	81.8	93.7	66.6	84.09

Figure 4.4: Success rate of ARC-AC with an equilibrated split

Results of the mammogram classification with ARC-AC were presented in [3].

ARC-BC

The following subsection presents the results when ARC-BC was employed with the new database organization presented above. For Table 4.3 that is presented below, the association rules are discovered setting a starting minimum support at 25% and the minimum confidence at 50%. The computation

of the actual support with which the database is mined is computed in an adaptive way. Starting with the given minimum support the dataset is mined, then a set of association rules is found. These rules are ordered and used as a classifier to test the classifier on the training set. When the accuracy on the training set is higher than a given accuracy threshold, the mining process is stopped, otherwise the support is decreased ($\sigma = \sigma - 1$) and the process is continued. As a result, different classes are mined at different supports. The parameters in the tests with the results below are: minimum support 25%, minimum confidence 50% and the accuracy threshold is 95%.

	1st rule		ordered		cut rules		remove specific	
Split	#rules	accuracy	#rules	accuracy	#rules	accuracy	#rules	accuracy
1	22	76.67	1121	80.00	856	76.67	51	60.00
2	18	86.67	974	93.33	755	90.00	48	86.67
3	22	83.33	823	86.67	656	86.67	50	76.67
4	22	63.33	1101	76.67	842	66.67	51	53.33
5	33	56.67	1893	70.00	1235	70.00	63	50.00
6	16	66.67	1180	76.67	958	73.33	51	63.33
7	30	66.67	1372	83.33	1055	73.33	58	53.33
8	26	66.67	1386	76.67	1089	80.00	57	46.67
9	20	66.67	1353	76.67	1130	76.67	52	60.00
10	18	76.67	895	83.33	702	80.00	51	76.67
avg(%)	22.7	**71.02**	1209.8	**80.33**	927.8	**77.33**	53.2	**62.67**

Table 4.3: Classification accuracy over the 10 splits using ARC-BC

Classification in the first two columns of Table 4.3 was done by assigning the image to the category attached to the first rule (the one with the highest confidence) that applies to the test image (see Table 4.3 columns under '1st rule'). However, pruning techniques are employed before so that a high quality set of rules is selected. The pruning technique used in this case is a modified version of the database coverage (i.e. selecting a set of rules that classifies most transactions presented in the training set). Given a set of rules, the main idea is to find the best rules that would make a good distinction between the classes. The given set of rules is ordered. Take one rule at a time and classify the training set for each class. If the consequent of the rule indicates class c_i keep that rule, only if it correctly classifies some objects in c_i training

set and doesn't classify any in the other classes. The transactions that were classified are removed from the training set.

The next columns in Table 4.3 are results of classification that uses the most powerful class in the set of rules. The difference is as follows: in the first two columns the set of rules that form the classifier is the set of rules extracted at the mining stage but ordered according to the confidence and support of the rules (support was normalized so that the ordering is possible even if the association rules are found by category)(see Table 4.3 columns under 'ordered'); in the next two columns after the rules were ordered the conflicting rules (see Section 3.2.3) were removed (see Table 4.3 columns under 'cut rules'); in the last two columns (see Table 4.3 columns under 'remove specific') from the ordered set of rules the specific ones were removed if they had lower confidence (see Section 3.2.1).

The best performance was obtained when no pruning was done on the obtained set of rules. When the specific rules were removed the accuracy went down by more than 15%. However, we have to notice the difference in the number of rules in each case. Although, the accuracy went down when the specific rules were removed the number of rules in the classifier reduced considerably. This is an important fact due to understability of rules. From a reduced set of rules, a physician may use only some of them in the medical decision that has to made. In the case of manual tuning of rules it is desirable to have small number of rules. In conclusion, although, the automatic classifier gave bad results when specific rules were removed, this set of rules could be used in a semi-automatic system.

We also present precision/recall graphs in Figure 4.5 to show that both false positive and false negative are very small, which means that for abnormal images was a very small number of false negative which is very desirable in medical image classification.

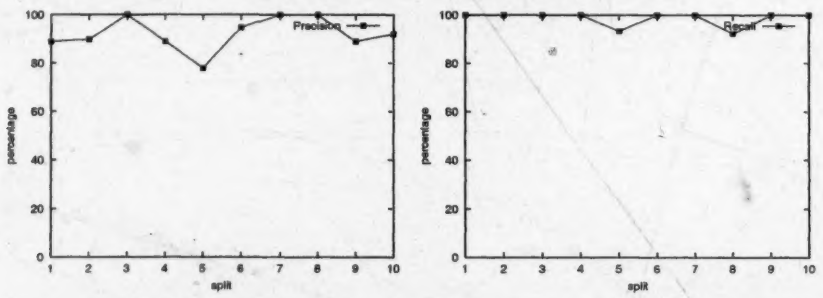
The formulas for precision and recall are given below:

$$R = \frac{TP}{TP + FN} \tag{4.4}$$

$$P = \frac{TP}{TP + FP} \tag{4.5}$$

The terms used to express precision and recall are given in the contingency table Table 4.4, where TP stands for true positives, FP for false positives, FN for false negatives and TN for true negatives.

From the graphs presented in Figure 4.5 one can observe that for both precision and recall for normal cases the values are very high. In addition, we can notice from equations 4.4 and 4.5 that the values for FP and FN tend to zero when precision and recall tend to 100%. Thus, the false positives and in particular false negatives are almost null with our approach.



(a) Precision (b) Recall

Figure 4.5: (a)Precision over the ten splits ; (b) Recall over the ten splits;

Category *cat*		human assignments	
		Yes	No
classifier assignments	Yes	TP	FP
	No	FN	TN

Table 4.4: Contingency table for category *cat*

In Table 4.5 the classification is done using the association rules obtained when mining the entire dataset at once as in [3]. However, the transactional database was organized as explained in Section 4.2.4. In the first two columns the set of rules that form the classifier is the set of rules extracted at the mining stage but ordered according to the confidence and support of the rules (see Table 4.5 columns under 'ordered'); in the next two columns after the

rules were ordered the conflicting rules (see Section 4.2.4) were removed (see Table 4.5 columns under 'cut rules').

Split	ordered		cut rules	
	#rules	accuracy	#rules	accuracy
1	6967	53.33	6090	53.33
2	5633	86.67	4772	86.67
3	5223	76.67	4379	76.67
4	6882	53.33	5938	53.33
5	7783	50.00	6878	50.00
6	7779	60.00	6889	60.00
7	7120	46.67	6209	46.67
8	7241	43.33	6364	43.33
9	7870	53.33	6969	53.33
10	5806	76.67	4980	76.67
avg(%)	6830.4	**60.00**	5946.8	**60.00**

Table 4.5: Classification accuracy over the 10 splits using ARC-AC [3]

As observed from the two tables presented above, the accuracy reached when ARC-BC is used is higher than the one obtained when the training set was mined at once with ARC-AC. However, the accuracy reached in [3] with ARC-AC was actually higher than in this case (69.11%). These results prove the importance of choosing the right data cleaning technique and data organization in reaching an effective and efficient data mining system.

Not only in accuracy does ARC-BC outperform ARC-AC, but in time measurements as well (41.315 seconds versus 199.325 seconds for training and testing for all ten splits). All tests were performed on an AMD Athlon 1.8 GHz.

4.3 Gene Categorization

With the new advances in the working on genes, there are now experimental methods that allow biologists to measure some aspect of cellular "activity" for thousands of genes or proteins at a time. The problem that arises when such experiments are done is how to deal with all this amount of data. Data mining techniques could perform well on large amounts of data. That is the motiva-

tion for trying our associative classifier on gene categorization. The main idea would be to capture some patterns that can be used later in the classification process. Given a training set containing various features describing the genes and their class label attached the task is to discover some patterns (i.e. association rules, having as antecedent a set of features and as consequent a class label) to be used in the categorization process when new genes are provided.

This task was one of the problems raised at the KDD Cup 2002 [1]. The training and test data that were provided came from some recent biological experiments. In the training set for each gene was provided the target class as well as various features (i.e. gene localization and function) associated with the gene. Along with the features provided, for each gene a set of abstracts from the scientific literature (MEDLINE) was given. The goal of the task was to find interesting patterns in the data that could built a good prediction model.

The difficulty of the task consisted in the unbalanced data with respect to the target classes (one big class had 97% of the training set, while other two had the rest). In addition, the feature set was very sparse, which made the task even more challenging. That was the motivation for trying our ARC-BC algorithm on this data set. Our classification system was able to better distinguish among classes, by allowing to find some interesting patterns in the small classes as well. We used our associative classifier not only on the overall classification task, but as well on mining the abstracts to fill in some missing attributes in the set of features.

On this dataset we applied the ARC-BC algorithm due to the huge difference in the class distribution. Not only the classification task was difficult due to imbalanced data, but the evaluation as well. A classical evaluation measure in classification is the accuracy. However, accuracy was not an appropriate measure in this classification case. That is why the evaluation chose for this task was ROC (relative operating curve) curve. The ROC curve records various combinations between the false positives and false negatives. That is why is more suitable for rare classes. The performance of the system is measured by the area under this curve. Our algorithm proved to perform well in comparison

with the other competitors.

Chapter 5

Experimental Results: multiple label classification

In this chapter we introduce the experimental results obtained when our approaches were tested on text collections. These collections are particular in the sense that classes overlap. This means that a document does not only belong to one class only, but can fall into many classes, the classification performed is multiple-label. The classification performed is called multi-label classification.

5.1 Benchmark Collections

5.1.1 Text Corpora

Database Description

In order to be able to objectively evaluate our algorithm vis-à-vis other approaches, like other researchers in the field of automatic text categorization, we used the Reuters-21578 text collection [42] and OHSUMED collection as benchmarks. These two databases are described below. Text collections for experiments are usually split into two parts: one part for training or building the classifier and a second part, for testing the effectiveness of the system.

There are many splits of the Reuters collection; we chose to use the *ModApte* version. This split leads to a corpus of 12,202 documents consisting of 9,603 training documents and 3,299 testing documents. There are 135 topics to which documents are assigned. However, only 93 of them have more than one document in the training set and 82 of the categories have less than 100

documents [22]. Obviously, the performances in the categories with just a few documents would be very low, especially for those that do not even have a document in the training set. Among the documents there are some that have no topic assigned to them. We chose to ignore such documents since no knowledge can be derived from them. Finally we decided to test our classifiers on the ten most populated categories with the largest number of documents assigned to them in the training set. Other researchers have used the same strategy [45], which constrained us to do the same for the sake of comparison. By retaining only the ten most populated categories we have 6488 training documents and 2545 testing documents. On these documents we performed stopword elimination but no stemming.

The other collection used in our experiments is the OHSUMED text collection, which was compiled by William Hersh. This collection was developed at the Oregon Health Science University and it is available online at ftp://medir.ohsu.edu/pub/ohsumed. With a corpus of 348,543 records that have MeSH (Medical Subject Headings) categories assigned, it consists of Medline records from the years 1987 to 1991. MeSH categories represent human-assigned terms to each record from the (MeSH) vocabulary. From the 348,543 records, only 233,445 of them have abstracts. We used in our experiments only those 233,445 records that have both title and abstract. We conducted two major experiments on this data collection. First, we used all the 233,445 documents. The training/testing split used was the same as that reported by other researchers in the literature: the documents from the first four years (183,229) were used for training, while those from 1991 (50,216) are considered for testing. For comparison reasons, we focused in this study on the Heart Disease sub-tree of the Cardiovascular Diseases tree as other researchers reported in their work [5, 10, 18]. The Heart Disease sub-tree contains 119 categories, but due to the variance in frequency of these categories, we used only those categories that had at least 75 documents per category. This new pruning led us to the use of 49 categories out of 119.

In this thesis, for the experiments performed, we concentrated more on the Reuters dataset. As a result, more detailed results are given for this collection.

Preprocessing Techniques

In our approach, we model text documents as transactions where items are words from the document as well as the categories to which the document belongs, as described prior. A data cleaning phase is required to weed out those words that are of no interest in building the associative classifier. We consider stemming and stopwording as well as the transformation of documents into transactions as a pre-processing phase. Stopword removal is done according to the term frequency values and a given list of stopwords. We have opted to selectively turn stemming, as well as stopwording for that matter, on and off depending upon the data set to categorize. Figure 5.1 illustrates the phases of the association-rule-based text categorizer construction with the optional stemming and stopwording modules. Stopwords removal considerably reduces the dimensions of the transactional database and thus significantly improves the rule extraction time (i.e. training time). Moreover, while we use a common stopword list in English [22], too frequent terms that are associated to all categories can be automatically added as words to reject. Note that the stopword lists from any language can be used as well. It is only after the terms are selected from the cleansed documents that the transactions are formed. The subsequent phase consists of discovering association rules from the set of cleansed transactions.

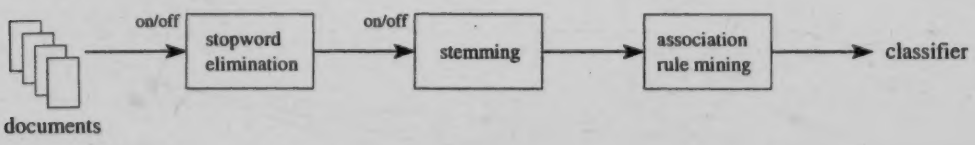


Figure 5.1: Classifier

5.2 Evaluation Measures

A classical evaluation measure in classification is the accuracy. However, accuracy is not an appropriate measure in all classification cases. There are two well-known such cases. One of them is in the case of rare classes and the other one appears when multi-label classification is performed. In our experiments,

we encounter both discussed cases. That is why we need other evaluation measures.

Several measurements have been used in previous studies for evaluating classification methods. Some measures, as well as those used in our evaluation, can be defined in terms of precision and recall. The formulas for precision and recall are expressed in the previous chapter in Formulae 4.4 and 4.5. The terms used to express precision and recall are given in the contingency table Table 4.4.

For evaluating the effectiveness of our system, we used the F_1 measure and breakeven points. F1 measure is a particular case of the F_β (Equation 5.1) measure.F_β measure is a common measure used in information retrieval and it can be also expresses as $F_\beta \equiv 1 - E_\beta$ where E_β is the effectiveness measure introduced by van Rijsbergen in 1979 [49]. F-measure is a combination of precision (P) and recall (R) and has the following formula:

$$F_\beta = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R} \tag{5.1}$$

F1 measure is obtained when β equals 1. The breakeven point (Equation 5.2) is the point at which precision equals recall and it is obtained as reported in [6].

$$BEP = \frac{R + P}{2} \tag{5.2}$$

When dealing with multiple classes there are two possible ways of averaging these measures, namely, macro-average and micro-average. In the macro-averaging, one contingency table as Table 4.4 per class is used, the performance measures are computed on each of them and then averaged. In micro-averaging only one contingency table is used, an average of all the classes is computed for each cell and the performance measures are obtained therein. The macro-average weights equally all the classes, regardless of how many documents belong to it. The micro-average weights equally all the documents, thus favouring the performance on common classes. We tested on the data collections both algorithms that we presented in this work.

5.3 Experimental Results

Firstly, we tested the ARC-AC algorithm on the Reuters data set. However, as expected on unbalanced data, from the empirical analysis conducted in Chapter 3, the performance on rare classes was very poor. Although the overall accuracy was about 70% on average over the ten classes, 5 out of 10 classes were not predicted at all. As it can be seen in Table 5.1 there are two big classes in the news dataset and 8 out of 10 with less 550 instances. Because the performance of the ARC-AC algorithm was so low, all the results that are reported in this chapter are for ARC-BC.

Category	training set	testing set
acq	1650	719
corn	182	56
crude	389	189
earn	2878	1087
grain	433	149
interest	347	131
money-fx	538	179
ship	196	89
trade	370	117
wheat	212	71

Table 5.1: Statistics for ten most populated Reuters categories

In Table 5.2 we report the micro-averages for ARC-BC when both support and dominance factor were varied. The results are computed on the ten most populated categories in Reuters dataset. As described in Chapter 3 we applied some pruning techniques on the discovered association rule set. Table 5.2 reports micro-averages when the classifier is built by employing the pruning methods (i.e. no pruning at all (w/o pruning), removing specific rules (rm-s) and removing specific rules plus database coverage (rm-s + db-cov) applied). It can be observed that the best performance is obtained when no pruning technique is applied. However, the number of rules in the classifier when the pruning techniques are applied is much smaller. Here it is a trade-off between the performance and the time. In addition, performance could be improved with manual tunning when pruning techniques are applied. A human

expert can read and change hundreds of rules to improve the performance when pruning is done, while reading thousands of rules without pruning is not feasible.

micro BEP	supp=10%			supp=15%			supp=20%		
	δ=50	δ=70	δ=90	δ=50	δ=70	δ=90	δ=50	δ=70	δ=90
w/o pruning	82.0	84.6	85.6	81.8	84.4	85.8	81.0	86.3	84.0
rm-s	65.5	72.9	76.3	68.2	75.0	78.8	68.2	76.1	79.7
rm-s + db-cov	71.0	79.0	82.3	71.4	79.6	73.1	70.1	70.4	84.1

Table 5.2: Precision/Recall-breakeven point micro-averages for ARC-BC

BEP	ARC-BC with δ=50			Bayes	Rocchio	C4.5	k-NN	bigrams	SVM (poly)	SVM (rbf)
	10%	15%	20%							
acq	**90.9**	89.9	87.8	91.5	92.1	85.3	92.0	73.2	94.5	95.2
corn	69.6	**82.3**	70.9	47.3	62.2	87.7	77.9	60.1	85.4	85.2
crude	77.9	77.0	**80.7**	81.0	81.5	75.5	85.7	79.6	87.7	88.7
earn	**92.8**	89.2	86.6	95.9	96.1	96.1	97.3	83.7	98.3	98.4
grain	68.8	72.1	**73.1**	72.5	79.5	89.1	82.2	78.2	91.6	91.8
interest	70.5	70.1	**75.3**	58.0	72.5	49.1	74.0	69.6	70.0	75.4
money-fx	70.5	**72.4**	70.5	62.9	67.6	69.4	78.2	64.2	73.1	75.4
ship	**73.6**	73.2	63.0	78.7	83.1	80.9	79.2	69.2	85.1	86.6
trade	68.0	69.7	**69.8**	50.0	77.4	59.2	77.4	51.9	75.1	77.3
wheat	84.8	**86.5**	85.3	60.6	79.4	85.5	76.6	69.9	84.5	85.7
micro-avg	**82.1**	81.8	81.1	72.0	79.9	79.4	82.3	73.3	85.4	86.3
macro-avg	76.74	**78.24**	76.32	65.21	79.14	77.78	82.05	67.07	84.58	86.01

Table 5.3: Precision/Recall-breakeven point on ten most populated Reuters categories for ARC-BC and most known classifiers

Tables 5.3 and 5.4 (results for the other methods asreported in [45]) show a comparison between our ARC-BC classifier and other well-known methods. The measures used are precision/recall-breakeven point, micro-average and macro-average on ten most populated Reuters categories. Our system proves to perform well as compared to the other methods. It outperforms most of the conventional methods, it has similar performance with kNN but it does not perform better than SVM. In addition to these results, our system has two more features. First, it is very fast in both training and testing phases (see Table 5.7). The times reported are for all training and testing documents. Sec-

System	Type	Micro-BEP
	probabilistic	81.5
	probabilistic	72.0
	decision trees	88.4
C4.5	decision trees	79.4
FindSim	batch linear	64.6
Rocchio	batch linear	79.9
k-NN	example-based	82.3
	SVM	92.0
SVM Light	SVM	86.4
	Bayesian net	85.0
ARC-BC	association rules	86.3

Table 5.4: Micro-average Precision/Recall-breakeven point for ten most populated Reuters categories with different classifiers

net $\wedge$ profit $\Rightarrow$ earn
agriculture $\wedge$ department $\wedge$ grain $\Rightarrow$ corn
assistance $\wedge$ bank $\wedge$ england $\wedge$ market $\wedge$ money $\Rightarrow$ interest
acute $\wedge$ coronary $\wedge$ function $\wedge$ left $\wedge$ ventricular $\Rightarrow$ myocardial-infarction
ambulatory $\wedge$ ischemia $\wedge$ myocardial $\Rightarrow$ coronary-disease
antiarrhythmic $\wedge$ effects $\Rightarrow$ arrhythmia

Table 5.5: Examples of association rules composing the classifier.

ond, it produces readable and understandable rules that can be easily modified by humans (see Table 5.5).

Table 5.6 reports the improvements in the response of the system when human tuning was applied. The support was set to 20% which made the classifier to perform very poor on *corn* and *wheat* categories . By reading the rules we noticed that by adding 4 more rules for each of these categories the performance improved as presented in Table 5.6.

A comparison between the pruning methods is given in Table 5.8. By applying the pruning methods, the accuracy of the classifier is not improved. However, the reduction in number of rules represents a step further in manually or automatically tuning of the system.

With the OHSUMED collection, our algorithm ARC-BC outperformed the Rocchio classifier (29%) and equaled the neural network classifiers at 40% as

ARC-BC supp=20% δ=90 rm-s+db-cov		
BEP	initial set of rules	manual tuned set of rules
acq	89.6	89.6
corn	2.0	63.6
crude	80.0	80.0
earn	92.7	92.7
grain	92.5	81.9
interest	57.7	57.7
money-fx	77.9	77.9
ship	61.3	61.3
trade	75.5	75.5
wheat	6.0	63.5
micro-avg	84.14	84.62
macro-avg	63.55	74.41

Table 5.6: Micro-average Precision/Recall-breakeven point for ten most populated Reuters categories - manual tuning of the classifier

support	training	testing
10%	18	3
15%	9	2
20%	8	2

Table 5.7: Training and testing time (in seconds) with respect to the support threshold for Reuters-21578 dataset

reported in the literature [43]. However, the results were lower than exponentiated gradient (EG) algorithm (50%) and Widrow-Hoff algorithm (55%) also reported in [43].

Some preliminary results were presented in [55]. However, this chapter gives a more detailed and thorough description of the results obtained when our approach was tested on text collections.

BEP	ARC-BC with $\delta = 50$ and supp=15%		
	w/o pruning	rm-s	rm-s + db-cov
No rules	**3072**	**383**	**127**
acq	89.9	84.2	76.6
corn	82.3	62.7	2.8
crude	77.0	58.4	64.8
earn	89.2	85.6	78.0
grain	72.1	56.4	71.8
interest	70.1	60.8	63.6
money-fx	72.4	62.3	68.7
ship	73.2	67.6	59.0
trade	69.7	59.2	73.5
wheat	86.5	46.9	24.7
micro-avg	81.8	68.2	71.4
macro-avg	78.24	64.40	58.53

Table 5.8: Precision/Recall-breakeven point for ten most populated Reuters categories with different pruning methods

Chapter 6

Conclusions

In this thesis we proposed two classification algorithms, *Association Rule-based Classification with All Categories* and *Association Rule-based Classification by Category*. This chapter summarizes the approach proposed, its contributions and some future directions.

6.1 Thesis Summary

In this work we presented two new classification methods based on association rule mining. We introduced the association rules theoretical background. Then, we defined our approaches in building a classification system. Finally, we discussed some experimental results performed on image and text collections.

The contribution of this thesis is a new classification approach, using association rules which has the following advantages:

- there is no assumption of term independence in our method. It proved to perform as well as other proposed methods, if not better;
- it is fast during both training and categorization phases;
- the system is able to perform both, single-label and multi-label classification;
- it can be applied on any transactional database, regardless of its content;
- can handle large dimensional spaces (i.e. long transactions);

- the classification model built using our approach can be read, understood and modified by humans.

6.2 Conclusions

Classification and association rules are two important tasks in Knowledge Discovery in Data. In this thesis we took advantage of the constrained association rules to propose a classification solution. We introduced new approaches to develop efficient classification systems and we tested them on various data sets. We compared our algorithm with other two associative classifiers and the state-of-the-art C4.5 in Chapter 4. Then, in the same chapter we used both approaches to classify mammograms. The algorithms performed well and such a system could be a useful tool in assisting medical staff in their decisions. In addition, to single-label classification we tested our system on a gene database. Chapter 5 presented the results obtained when multi-label classification was performed on text collections. Although the system achieved good results and it is comparable with others in the literature, there is still room for improvement.

6.3 Future Work

This section discusses some ideas that could lead to future improvements for the system proposed in this thesis. Some ideas would be the following:

- Currently the discovered rules consider only the presence of features for classification. One possibility for improving the classification quality would be to take into account the absence of terms in the classification rules as well. Absence of features can indeed discriminate among classes.

- When text categorization is employed, more work in the pre-processing phase could be done. Right now, we eliminate the stopwords, which reduce considerably the term space. However, some other techniques could be used to reduce the space dimensionality. Term pruning accord-

ing to TF/IDF (term frequency/inverse document frequency) or latent semantic indexing could be used.

- Another possibility that could influence the classification stage would be to attach weights to the association rules according to their importance.
- In addition, taking into account the re-occurence of terms in a document as a possible discriminator could lead to improvement in performance.

Bibliography

[1] Kdd cup 2002 task 2. http://www.biostat.wisc.edu/~craven/kddcup/, 2002.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data*, pages 207–216, Washington, D.C., May 1993.

[3] Maria-Luiza Antonie, Osmar R. Zaïane, and Alexandru Coman. Application of data mining techniques for medical image classification. In *In Proc. of Second Intl. Workshop on Multimedia Data Mining (MDM/KDD'2001) in conjunction with Seventh ACM SIGKDD*, pages 94–101, San Francisco, USA, 2001.

[4] C. Apte, F.J. Damerau, and S.M. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.

[5] Chidanand Apté and Sholom M. Weiss. Data mining with decision trees and decision rules. *Future Generation Computer Systems*, 13:197–210, 1997.

[6] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. On feature distributional clustering for text categorization. In *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 146–153, New Orleans, US, 2001.

[7] C.L. Blake and C.J. Merz. Uci repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[8] D. Brazokovic and M. Neskovic. Mammogram screening using multiresolution-based image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(6):1437–1460, 1993.

[9] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[10] C. Chen and G. Lee. Image segmentation using multiresolution wavelet analysis and expectation-maximization (em) algorithm for digital mammography. *International Journal of Imaging Systems and Technology*, 8(5):491–504, 1997.

[11] Peter Clark and Tim Niblett. The cn2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

[12] W.W. Cohen and H. Hirsch. Joins that generalize: text classification using whirl. In *4th International Conference on Knowledge Discovery and Data Mining (SigKDD'98)*, pages 169–173, New York City,USA, 1998.

[13] W.W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141 –173, 1999.

[14] W. Bruce Croft, Howard R. Turtle, and David D. Lewis. The use of phrases and structured queries in information retrieval. In Abraham Bookstein, Yves Chiaramella, Gerard Salton, and Vijay V. Raghavan, editors, *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Chicago, Illinois, USA, October 13-16, 1991 (Special Issue of the SIGIR Forum)*, pages 32–45. ACM, 1991.

[15] A. Dhawan et al. Radial-basis-function-based classification of mammographic microcalcifications using texture features. In *Proc. of the 17th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 535–536, 1995.

[16] H. Chan et al. Computerized analysis of mammographic microcalcifications in morphological and feature spaces. *Medical Physics*, 25(10):2007–2019, 1998.

[17] H. Li et al. Marcov random field for tumor detection in digital mammography. *IEEE Trans. Medical Imaging*, 14(3):565–576, 1995.

[18] H. Li et al. Fractal modeling and segmentation for the enhancement of microcalcifications in digital mammograms. *IEEE Trans. Medical Imaging*, 16(6):785–798, 1997.

[19] I. Christoyianni et al. Fast detection of masses in computer-aided mammography. *IEEE Signal Processing Magazine*, pages 54–64, Jan 2000.

[20] I. Christoyianni et al. Fast detection of masses in computer-aided mammography. *IEEE Signal Processing Magazine*, pages 54–64, 2000.

[21] U.M. Fayyad, G. Piatesky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI Press and The MIT Press, CA, USA, 1996.

[22] C. Fox. *Information Retrieval: Data Structures and Algorithms*, chapter Lexical analysis and stoplists, pages 113–116. Prentice-Hall, 1992. ftp://sunsite.dcc.uchile.cl/pub/users/rbaeza/irbook/stopper/.

[23] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. In Jude W. Shavlik, editor, *Proceedings of ICML-98, 15th International Conference on Machine Learning*, pages 170–178, Madison, US, 1998. Morgan Kaufmann Publishers, San Francisco, US.

[24] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[25] N. Fuhr and C. Buckley. A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems*, 9(3):223–248, 1991.

[26] Rafael C. Gonzalez and Richard. E. Woods. *Digital Image Processing.* Addison-Wesley, 1993. second edition.

[27] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM-SIGMOD*, Dallas, 2000.

[28] Jiawei Han and Micheline Kamber. *Data Mining, Concepts and Techniques.* Morgan Kaufmann, 2001.

[29] D. A. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *17th ACM International Conference on Research and Development in Information Retrieval (SIGIR-94)*, pages 282–289, 1994.

[30] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *10th European Conference on Machine Learning (ECML-98)*, pages 137–142, 1998.

[31] S. Lai, X. Li, and W. Bischof. On techniques for detecting circuscribed masses in mammograms. *IEEE Trans. Medical Imaging*, pages 377–386, 1989.

[32] D. Lewis. Naïve (bayes) at forty: The independence assumption in information retrieval. In *10th European Conference on Machine Learning (ECML-98)*, pages 4–15, 1998.

[33] David D. Lewis and W. Bruce Croft. Term clustering of syntactic phrases. In Jean-Luc Vidick, editor, *SIGIR'90, 13th International Conference on Research and Development in Information Retrieval, Brussels, Belgium, 5-7 September 1990, Proceedings*, pages 385–404. ACM, 1990.

[34] H. Li and K. Yamanishi. Text classification using esc-based stochastic decision lists. In *8th ACM International Conference on Information and*

Knowledge Management(CIKM-99), pages 122 –130, Kansas City,USA, 1999.

[35] W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *IEEE International Conference on Data Mining (ICDM'01)*, San Jose, California, Novermber 29-December 2 2001.

[36] Y. H. Li and A. K. Jain. Classification of text documents. *The Computer Journal*, 41(8):537–546, 1998.

[37] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *ACM Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD'98)*, pages 80–86, New York City, NY, August 1998.

[38] M. Maron. Automatic indexing: An experimental inquiry. *Journal of the Association for Computing Machinery*, 8(3):404–417, 1961.

[39] http://www.wiau.man.ac.uk/services/MIAS/MIASweb.html.

[40] I. Moulinier and J.-G. Ganascia. Applying an existing machine learning algorithm to text categorization. In S.Wermter, E.Riloff, and G.Scheler, editors, *Connectionist statistical,and symbolic approaches to learning for natural language processing*. Springer Verlag, Heidelberg, Germany, 1996. Lecture Notes for Computer Science series, number 1040.

[41] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[42] The reuters-21578 text categorization test collection. http://www.research.att.com/~lewis/reuters21578.html.

[43] M.E. Ruiz and P. Srinivasan. Neural networks for text categorization. In *22nd ACM SIGIR International Conference on Information Retrieval*, pages 281–282, Berkeley, CA, USA, August 1999.

[44] Robert E. Schapire. Theoretical views of boosting. In Paul Fischer and Hans U. Simon, editors, *Proceedings of EuroCOLT-99, 4th European Conference on Computational Learning Theory*, pages 1–10, Nordkirchen, DE, 1999. Lecture Notes in Computer Science, number 1572, Springer Verlag, Heidelberg, DE.

[45] F. Sebastiani. Machine learning in automated text categorization. Technical Report IEI-B4-31-1999, Consiglio Nazionale delle Ricerche, Pisa, Italy, 1999.

[46] C. M. Tan, Y. F. Wang, and C. D. Lee. The use of bigrams to enhance text categorization. *Journal of Information Processing and Management*, 2002. http://www.cs.ucsb.edu/ yfwang/papers/ig&m.pdf.

[47] Uci knowledge discovery in databases archive. http://kdd.ics.uci.edu/.

[48] A. Vailaya, M.A.T. Figueiredo, A.K. Jain, and H-J. Zhang. Image classification for content-based indexing. *IEEE Trans. Image Processing*, 10(1):117–130, 2001.

[49] C.J. van Rijsbergen. *Information Retrieval.* Butterworths, London, 1979.

[50] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer Verlag, Heidelberg, DE, 1995.

[51] T. Wang and N. Karayiannis. Detection of microcalcification in digital mammograms using wavelets. *IEEE Trans. Medical Imaging*, pages 498–509, 1998.

[52] S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn.* Morgan Kaufmann, 1991.

[53] Y. Yang. An evaluation of statistical approaches to text categorization. Technical Report CMU-CS-97-127, Carnegie mellon University, April 1997.

[54] Y.Yang and X.Liu. A re-examination of text categorization methods. In *22nd ACM International Conference on Research and Development in Information Retrieval (SIGIR-99)*, pages 42–49, Berkeley,US, 1999.

[55] Osmar R. Zaïane and Maria-Luiza Antonie. Clasifying text documents by associating terms with text categories. In *In Proc. of the Thirteenth Australasian Database Conference (ADC'02)*, pages 215–222, Melbourne, Australia, 2002.